ISSN 1818-1015
eISSN 2313-5417

2020 Volume 27 No 1

MODELING AND ANALYSIS OF INFORMATION SYSTEMS

SCIENTIFIC JOURNAL

Start date of publication — 1999
Published quarterly

FOUNDER
P.G. Demidov Yaroslavl State University

EDITORIAL OFFICE
14 Sovetskaya str., Yaroslavl 150003, Russian Federation

Website: http://mais-journal.ru
E-mail: mais@uniyar.ac.ru
Phone: +7 (4852) 79-77-73

2020 Том 27 № 1

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

НАУЧНЫЙ ЖУРНАЛ

Издается с 1999 года
Выходит 4 раза в год

УЧРЕДИТЕЛЬ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Ярославский государственный университет им. П. Г. Демидова»

РЕДАКЦИЯ
ул. Советская, 14, Ярославль, 150003, Российская Федерация
Website: http://mais-journal.ru
E-mail: mais@uniyar.ac.ru
Телефон: +7 (4852) 79-77-73

Свидетельство о регистрации СМИ ПИ № ФС 77–66186 от 20.06.2016 выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Подписной индекс – 31907 в Объединенном каталоге «Пресса России». Технический редактор, компьютерная вёрстка – М. С. Каряева. Подписано в печать 16.03.2020. Дата выхода в свет 31.03.2020. Формат 200×265 мм. Объем 132 с. Тираж 46 экз. Свободная цена. Заказ 010/020. Адрес типографии: ул. Советская, 14, оф. 109, Ярославль, 150003, Россия. Адрес издателя: Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003, Россия.

12+

Editor-in-Chief

Valery A. Sokolov . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Deputies Editor-in-Chief

Sergey D. Glyzin . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Eugeniy A. Timofeev . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Editorial Board Secretary

Egor V. Kuzmin . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

The Editorial Board

Sergei M. Abramov . . . . . Professor, Doctor of Sciences, Corresponding Member of Russian Academy of Sciences, Program Systems Institute of RAS (Pereslavl-Zalesskiy, Russia)
Lilian Aveneau . . . . . Professor, XLIM Laboratory, University of Poitiers (Poitiers, France)
Thomas Baar . . . . . Professor, Doctor, Hochschule für Technik und Wirtschaft Berlin, University of Applied Sciences (Berlin, Germany)
Olga L. Bandman . . . . . Professor, Doctor of Sciences, Supercomputer Software Department, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)
Vladimir N. Belykh . . . . . Professor, Doctor of Sciences, Volga State Academy of Water Transport (Nizhny Novgorod, Russia)
Vladimir A. Bondarenko . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Richard R. Brooks . . . . . Professor, Clemson University (South Carolina, USA)
Alex Dekhtyar . . . . . Professor, California Polytechnic State University (Cal Poly, California, USA)
Mikhail Dmitriev . . . . . Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)
Vladimir L. Dolnikov . . . . . Doctor of Sciences, Moscow Institute of Physics and Technology (Moscow, Russia)
Valery G. Durnev . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Yuri G. Karpov . . . . . Professor, Doctor of Sciences, St-Petersburg State Polytechnical University (Russia)
Sergey A. Kashchenko . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Lev S. Kazarin . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Andrei Yu. Kolesov . . . . . Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)
Nikolai A. Kudryashov . . . . Professor, Doctor of Sciences, MEPhI (Russia)
Olga Kouchnarenko . . . . . Professor at the Burgundy-Franche-Comte University, The FEMTO-ST Institute (CNRS 6174) (Besancon, France)
Irina A. Lomazova . . . . . Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)
George G. Malinetskiy . . . . Professor, Doctor of Sciences, M.V. Keldysh Institute of Applied Mathematics RAS (Moscow, Russia)
Victor E. Malyshkin . . . . . Professor, Doctor of Sciences, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)
Alexander V. Mikhailov . . . Professor, Doctor of Sciences, University of Leeds, School of Mathematics (Leeds, Great Britain)
Valery A. Nepomniaschy . . PhD, A.P. Ershov Institute of Informatics Systems SB RAS (Novosibirsk, Russia)
Nikolai Kh. Rozov . . . . . Professor, Doctor of Sciences, Lomonosov Moscow State University (Russia)
Philippe Schnoebelen . . . . . Senior Researcher, LSV, CNRS & ENS de Cachan (CACHAN, France)
Natalia Sidorova . . . . . Dr., Assistant Professor, Architecture of Information Systems group, Technische universiteit Eindhoven (Eindhoven, Netherlands)
Ruslan L. Smeliansky . . . . . Professor, Doctor of Sciences, Corresponding Member of RAS, Lomonosov Moscow State University (Russia)
Javid Taheri . . . . . Associate Professor, Ph.D., Karlstad University (Sweden)
Mark Trakhtenbrot . . . . . Dr., Holon Institute of Technology (Holon, Israel)
Dimitry Turaev . . . . . Professor of Applied Mathematics & Mathematical Physics, Imperial College (London, Great Britain)
Vladimir Zakharov . . . . . Doctor of Sciences, Professor, Lomonosov Moscow State University (Russia)

Главный редактор

В.А. Соколов . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Заместители главного редактора

С.Д. Глызин . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
Е.А. Тимофеев . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Ответственный секретарь

Е.В. Кузьмин . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Редакционная коллегия

С.М. Абрамов . . . . . д-р физ.-мат. наук, чл.-корр. РАН, Институт программных систем РАН им. А.К. Айламазяна (Россия)
L. Aveneau . . . . . проф., Университет Пуатье (Франция)
T. Baar . . . . . д-р наук, проф., Университет прикладных технических и экономических наук Берлина (Германия)
О.Л. Бандман . . . . . д-р техн. наук, Институт вычислительной математики и математической геофизики СО РАН (Россия)
В.Н. Белых . . . . . д-р физ.-мат. наук, проф., Волжская государственная академия водного транспорта (Россия)
В.А. Бондаренко . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
R. Brooks . . . . . проф., Университет Клемсона (США)
A. Dekhtyar . . . . . проф., Калифорнийский политехнический университет, департамент компьютерных наук (США)
М.Г. Дмитриев . . . . . д-р физ.-мат. наук, проф., ВШЭ (Россия)
В.Л. Дольников . . . . . д-р физ.-мат. наук, проф., МФТИ (Россия)
В.Г. Дурнев . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
В.А. Захаров . . . . . д-р физ.-мат. наук, проф., МГУ (Россия)
Л.С. Казарин . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
Ю.Г. Карпов . . . . . д-р техн. наук, проф., Санкт-Петербургский государственный технический университет (Россия)
С.А. Кащенко . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
А.Ю. Колесов . . . . . д-р физ.-мат. наук, проф., ЯрГУ (Россия)
Н.А. Кудряшов . . . . . д-р физ.-мат. наук, проф., Засл. деятель науки РФ, МИФИ (Россия)
O. Kouchnarenko . . . . . проф., Университет Бургундии - Франш-Комтэ (Франция)
И.А. Ломазова . . . . . д-р физ.-мат. наук, проф., ВШЭ (Россия)
Г.Г. Малинецкий . . . . . д-р физ.-мат. наук, проф., Институт прикладной математики им. М.В. Келдыша РАН (Россия)
В.Э. Малышкин . . . . . д-р техн. наук, проф., Институт вычислительной математики и математической геофизики СО РАН (Россия)
A. Mikhailov . . . . . д-р физ.-мат. наук, проф., Университет Лидса (Великобритания)
В.А. Непомнящий . . . . . канд. физ.-мат. наук, Институт систем информатики им. А.П. Ершова СО РАН (Россия)
Н.Х. Розов . . . . . д-р физ.-мат. наук, проф., чл.-корр. РАО, МГУ (Россия)
N. Sidorova . . . . . д-р наук, университет Эйндховена (Нидерланды)
Р.Л. Смелянский . . . . . д-р физ.-мат. наук, проф., член-корр. РАН, академик РАЕН, МГУ (Россия)
J. Taheri . . . . . доцент, Университет Карлстада (Швеция)
M. Trakhtenbrot . . . . . д-р комп. наук, Холонский технологический институт (Израиль)
D. Turaev . . . . . проф., Имперский колледж Лондона (Великобритания)
Ph. Schnoebelen . . . . . проф., Национальный центр научных исследований и Высшая нормальная школа Кашана (Франция)

Contents

Содержание

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

COMPUTER SYSTEM ORGANIZATION

Estimation of Length of Node-to-Node Paths Distribution in the Global Network

A. I. Kononova[1], A. V. Gorodilov[2]

DOI: 10.18255/1818-1015-2020-1-6-21

[1]National Research University of Electronic Technology, 1 Shokin sq., Moscow, Zelenograd 124498, Russia.
[2]Russkaya Moda (Russian fashion), 10 Rannyaya/Early str, Yaroslavl 150034, Russia.

MSC2020: 68M10
Research article
Full text in Russian

Received January 17, 2020
After revision February 25, 2020
Accepted February 28, 2020

The experiment aimed at finding a distribution of path lengths between nodes in the global network and an estimation of parameters of that distribution is described.In particular, the method of measurement of path length with traceroute utility of the GNU/Linux system and limitations on the selection of nodes imposed by traceroute are described. The measurement results are provided and high values of skewness and kurtosis for all resulting distributions are noted. Simulation model of this experiment was developed to test the experiment validity in the determination of distribution parameters in the global network. This model is also described. It is shown that high values of skewness and kurtosis of the measured distributions are not the result of the measurement technique, therefore the global network could not be described by the Barabási–Albert model. Several most viable hypotheses explaining differences in skewness and kurtosis of experimentally obtained path-length distribution estimations and values derived from the Barabási–Albert model are listed. Results of different hypotheses simulations are provided. It is shown that the most fitting hypothesis is that definitive influence on skewness and kurtosis of path-length distribution estimations is caused by the quasi pre-fractal structure of the global network.

Keywords: global network; routing; node-to-node distance distribution; experiment; Barabási–Albert model

INFORMATION ABOUT THE AUTHORS

Alexandra I. Kononova correspondence author	orcid.org/0000-0002-4178-3828. E-mail: illinc@bk.ru PhD.
Alexey V. Gorodilov	orcid.org/0000-0003-2887-8547. E-mail: kaverina@mail.ru PhD.

For citation: A. I. Kononova and A. V. Gorodilov, “Estimation of Length of Node-to-Node Paths Distribution in the Global Network”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 6-21, 2020.

© Kononova A. I., Gorodilov A. V., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

COMPUTER SYSTEM ORGANIZATION

К вопросу об оценках распределения длин путей между узлами в глобальной сети

А. И. Кононова[1], А. В. Городилов[2]

DOI: 10.18255/1818-1015-2020-1-6-21

[1]Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский институт электронной техники», пл. Шокина, 1, Москва, Зеленоград, 124498, Россия.
[2]Русская мода, ул. Ранняя, 10, Ярославль, 150034, Россия.

УДК 004.94
Научная статья
Полный текст на русском языке

Получена 17 января 2020 г.
После доработки 25 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

Описан эксперимент по оцениванию распределения длин путей между узлами в глобальной сети и его характеристик. В частности, показана методика измерения длины пути при помощи утилиты GNU/Linux traceroute и ограничения выбора узлов, налагаемые этим инструментом. Приведены результаты измерений, отмечены высокие значения асимметрии и эксцесса для всех полученных распределений. Описана имитационная модель эксперимента, разработанная для проверки корректности полученных оценок распределения длин путей между узлами в глобальной сети. Приведены результаты моделирования измерений. Показано, что высокие значения асимметрии и эксцесса измеренных распределений не обусловлены только методикой измерения, таким образом, глобальная сеть не описывается моделью Барабаши—Альберт. Перечислены основные гипотезы о причинах отличия асимметрии и эксцесса полученных экспериментально оценок распределения длин путей между узлами в глобальной сети от значений, соответствующих модели Барабаши—Альберт. Описаны результаты моделирования различных гипотез. Показано, что наиболее правдоподобной из них является предположение об определяющем влиянии квазипредфрактальной структуры глобальной сети на асимметрию и эксцесс оценок распределения длин путей между узлами.

Ключевые слова: глобальная сеть; маршрутизация; распределение длин путей; исследование структуры; безмасштабная модель Барабаши—Альберт

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александра Игоревна Кононова
автор для корреспонденции
orcid.org/0000-0002-4178-3828. E-mail: illinc@bk.ru
канд. техн. наук, доцент.

Алексей Владиславович Городилов
orcid.org/0000-0003-2887-8547. E-mail: kaverina@mail.ru
канд. техн. наук, доцент.

Для цитирования: A. I. Kononova and A. V. Gorodilov, “Estimation of Length of Node-to-Node Paths Distribution in the Global Network”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 6-21, 2020.

© Кононова А. И., Городилов А. В., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

В настоящее время передача данных через глобальную сеть используется, прямо или косвенно, практически во всех приложениях. В частности, для повышения качества передачи мультимедийных данных в приложении IP-телефонии в 2011 году разрабатывалась методика передачи данных с учётом особенностей сети [1].

Данные в рамках разработанной методики передаются через пиринговую сеть, построенную аналогично файлообменному протоколу BitTorrent, что позволило обеспечить обмен данных между узлами с серыми IP-адресами, прямая передача между которыми в рамках TCP/IP невозможна. Соответственно, каждый узел, участвующий в этой сети, хранит постоянно обновляемый список узлов-ретрансляторов (их адреса и характеристики). При этом количество хранимых узлов не должно быть как слишком малым (это необходимо для успешной передачи данных), так и слишком большим (хранение полной копии структуры сети в памяти каждого узла создаст не только высокую нагрузку на память узла, но и высокий уровень служебного трафика для поддержания актуальности этой копии). Кроме того, сам набор хранимых узлов не должен быть однородным — для повышения скорости информационного обмена бóльшая часть хранимых узлов должна физически находиться в сегментах сети, смежных с текущим, но для поддержания целостности сети каждый узел должен хранить в памяти несколько узлов, удалённых от текущего, причём не смежных между собой.

Для подбора [2] наилучшего размера и структуры списка хранимых узлов в 2011–2012 годах была начата серия экспериментов по оценке распределения длин путей (РДП) сетевого уровня модели TCP/IP глобальной сети при помощи программы traceroute. Изначально оценивалась только средняя длина пути; в дальнейшем был рассмотрен вопрос о прочих характеристиках РДП, а также о корректности полученных оценок.

Понятие длины пути между узлами

Рассмотрим процесс передачи данных по сети (будем рассматривать сетевой уровень четырёхуровневой модели TCP/IP). Пусть необходимо переслать пакет данных от узла α к узлу β. Тогда маршрут передачи данных, скорее всего, будет включать некоторое количество промежуточных узлов, служащих ретрансляторами пакета.

Определение 1. *Пусть маршрут передачи данных от узла α к узлу β, $\alpha \neq \beta$, включает k пересылок узел-узел (hops):*

$$\alpha \longrightarrow \alpha_1 \longrightarrow \ldots \longrightarrow \alpha_{k-1} \longrightarrow \beta \tag{1}$$

тогда будем называть число k длиной пути от узла α к узлу β *и обозначать $\lambda(\alpha, \beta)$:*

$$\lambda(\alpha, \beta) = k. \tag{2}$$

Также положим

$$\lambda(\alpha, \alpha) = 0. \tag{3}$$

При использовании стека протоколов TCP/IP маршрут передачи данных на сетевом уровне выбирается близким к оптимальному; но, поскольку разные промежуточные узлы оптимизируют разные параметры передачи, в общем случае маршрут передачи данных от узла β к узлу α отличается от маршрута передачи от α к β. Соответственно, возможна ситуация $\lambda(\alpha, \beta) \neq \lambda(\beta, \alpha)$. При этом сетевой уровень любой современной сети не содержит петель.

Кроме того, со временем глобальная сеть изменяет свою структуру, и как маршрут передачи данных, так и его длина $\lambda(\alpha, \beta)$ могут изменяться.

Определение 2. *Множество вершин (узлов) графа G будем обозначать как V_G, множество рёбер графа G — как E_G.*

Определение 3. *Под* распределением длин путей между узлами (РДП) *графа G будем понимать ряд чисел $\zeta(x)$, для каждой физически возможной длины пути x, то есть для каждого целого $x \geqslant 0$ показывающий, как часто длина x встречается среди всех возможных путей в G:*

$$\zeta(x) = p\big(\lambda(\alpha, \beta) = x \mid \alpha, \beta \in V_G\big). \tag{4}$$

В дальнейшем, чтобы подчеркнуть отличие распределения $\zeta(x)$ от его оценок, будем называть его полным РДП *графа G.*

Непосредственно измерить полное РДП для глобальной сети (то есть измерить и проанализировать длины $\lambda(\alpha, \beta)$ для всех возможных пар узлов α и β) невозможно как из-за её гигантских размеров (по различным оценкам [3] глобальная сеть содержит от 10^8 до 10^{10} узлов), так и из-за отсутствия в стеке TCP/IP средств для измерения расстояния между двумя произвольно взятыми узлами.

1. Измерение длины пути в GNU/Linux

Для исследования маршрутов сетевого уровня в GNU/Linux предназначена утилита traceroute. При запуске в командной строке узла α команды `traceroute` β результатом будет маршрут передачи данных $\alpha \longrightarrow \beta$, причём одна строка вывода traceroute соответствует одному узлу маршрута. Далее путём синтаксического анализа можно определить длину пути $\lambda(\alpha, \beta)$. Для этого использовались возможности оболочки Bash и утилиты GNU/Linux (в частности, wc, grep, sed), передача результатов измерений с различных корневых узлов осуществлялась при помощи svn.

Определение 4. *Будем далее называть* корневым узлом *узел α, на котором выполняется traceroute, и* оконечным узлом — *узел β, аргумент команды traceroute.*

Соответственно, описанную ниже схему оценивания РДП глобальной сети будем называть схемой корневой узел-оконечные узлы (КУОУ).

Таким образом, для измерения длины пути $\lambda(\alpha, \beta)$ необходимо иметь право на запуск программ на узле α (корневом) и знать IP-адрес или имя узла β (оконечного). Также необходима возможность соединения α с β, то есть оконечный узел β должен иметь белый IP-адрес либо находиться в одной подсети с α.

Ограничения на выбор оконечного узла гораздо мягче, чем для корневого. Вследствие этого количество возможных оконечных узлов может быть намного больше, чем корневых. В дальнейшем будем обозначать множество оконечных узлов $Q = \{\beta_1, \beta_2, \dots \beta_n\} \subseteq V_G$.

Соответственно, задавшись парой из корневого узла α и множества оконечных узлов Q и измеряя при помощи traceroute расстояния $\lambda(\alpha, \beta_j)$ от α до каждого из оконечных $\beta_j \in Q$, можно получить некоторую оценку РДП глобальной сети. Эта оценка зависит от выбора корневого узла, от множества оконечных узлов и от времени (от текущей конфигурации глобальной сети).

1.1. Результаты измерений

Для запуска программ было использовано три доступных корневых узла — домашние и рабочие компьютеры участников исследования (обозначаемые далее соответственно как A, B и C).

Также для оценивания РДП по методу КУОУ необходимо множество уникальных IP-адресов оконечных узлов. Так как полученные результаты планировалось использовать для организации передачи данных между компьютерами, аналогичными A, B, C, в качестве оконечных узлов рассматривались пользовательские компьютеры, а для серых IP-адресов — шлюзы их провайдеров. Подобные узлы участвуют в файлообмене по протоколу BitTorrent.

Изначально предполагалось, что характеристики РДП могут зависеть от географического положения оконечных узлов, что отчасти связано с используемым языком общения. Таким образом, было рассмотрено три крупных торрент-трекера, доступных на момент начала исследования и различающихся как по специфике распространяемого контента и языку общения, так и по физическому расположению большинства пользователей: pornolab.net, rutracker.org и thepiratebay.org.

Путём мониторинга IP-адресов их пользователей, принимающих участие в информационном обмене, были получены три множества уникальных IP-адресов размерами соответственно 564, 5222 и 540, обозначаемые далее как P, R и T. Эти множества затем и использовались как множества оконечных узлов.

От корневых узлов A и C было выполнено по четыре разнесённых во времени измерения расстояний до каждого доступного множества оконечных узлов; от узла B, из-за ограниченного времени доступа — только по одному измерению. Таким образом, всего было получено 27 оценок РДП, характеризуемых идентификатором i, составленным из множества оконечных узлов $Q \in \{P, R, T\}$, корневого узла $\alpha \in \{A, B, C\}$ и порядкового номера измерения для пары (α, Q):

$$i \in \{PA_0, PA_1, PA_2, PA_3, PB_0, PC_0, PC_1, PC_2, PC_3, RA_0, \dots RC_3, TA_0, \dots TC_3\} \tag{5}$$

Измерения с одним порядковым номером j из-за протяжённости процесса измерения (более суток), а также по техническим причинам, были выполнены не строго одновременно. Тем не менее, они выполнялись в сопоставимое время (в течение одного месяца). Измерения с порядковыми номерами j и $j + 1$ разделяет не менее восьми месяцев.

Результаты измерений представлены на рис. 1 и 2.

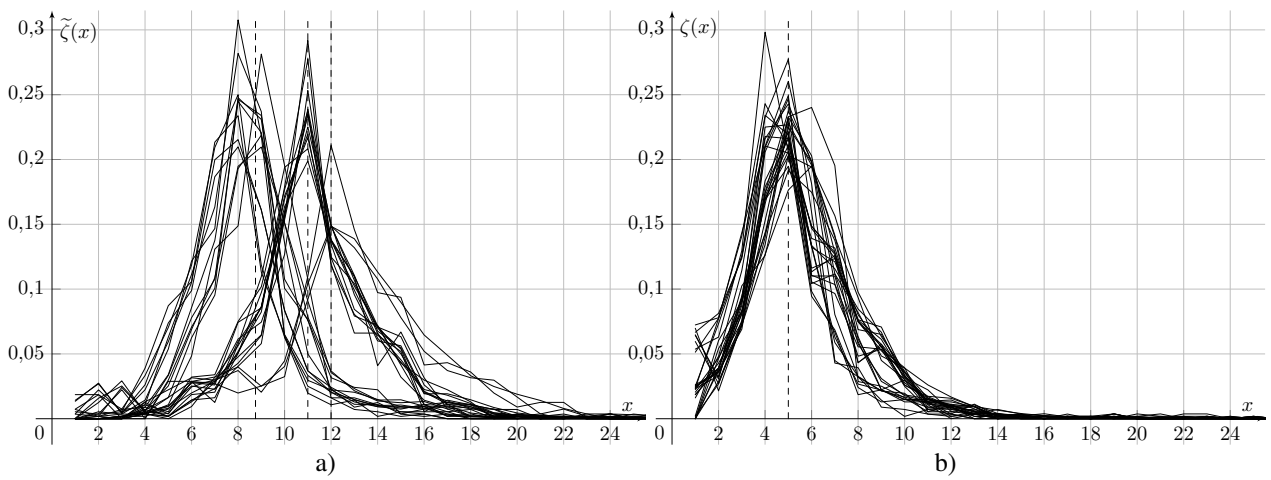


Fig. 1. Experimental evaluation of the NND distribution: a) including subnet of the root node provider, b) excluding subnet of the root node provider

Рис. 1. Экспериментально полученные оценки РДП: a) включая подсеть провайдера корневого узла, b) исключая подсеть провайдера корневого узла

На рис. 1 показано два варианта оценок РДП. Они были получены из следующих соображений. Все корневые узлы $\alpha \in \{A, B, C\}$ имели серые IP-адреса, а большинство оконечных $\beta \in Q$ — белые. Соответственно, маршрут передачи данных делился на две части: маршрут от α до шлюза его провайдера α_w (при этом все узлы $\alpha, \alpha_1, \dots \alpha_{w-1}$, находящиеся за шлюзом, имеют серые IP-адреса) и маршрут от шлюза провайдера α_w до β:

$$\alpha \longrightarrow \alpha_1 \longrightarrow \dots \alpha_{w-1} \longrightarrow \alpha_w \dots \longrightarrow \alpha_{k-1} \longrightarrow \beta \tag{6}$$

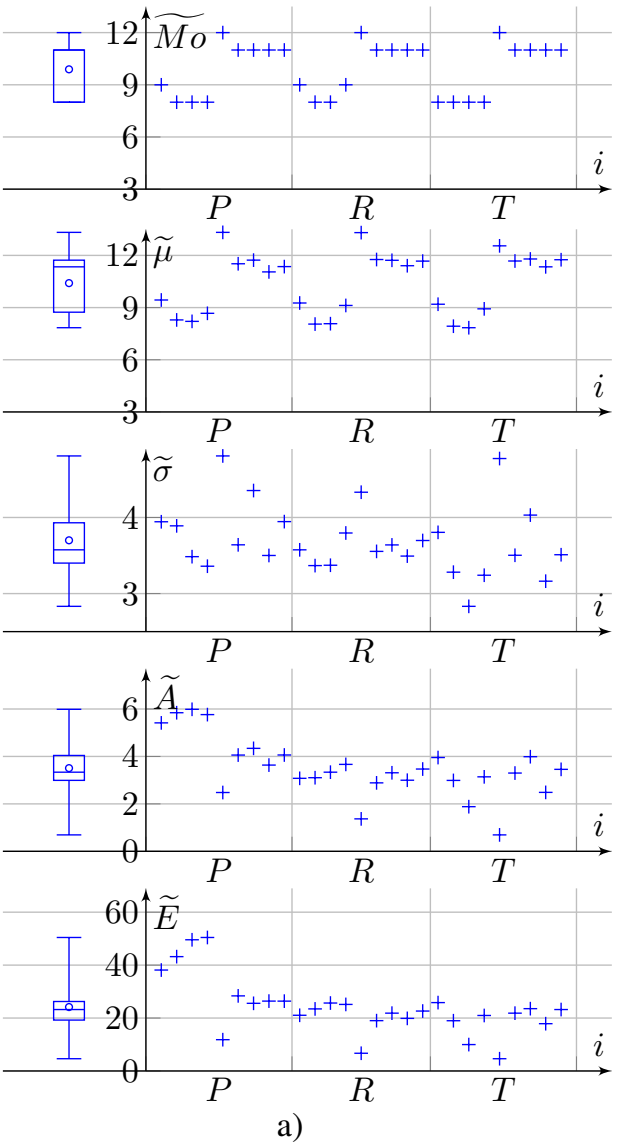


Fig. 2. Characteristics of experimentally obtained estimates of NND distribution: a) including a subnet of the root node provider, b) excluding a subnet of the root node provider

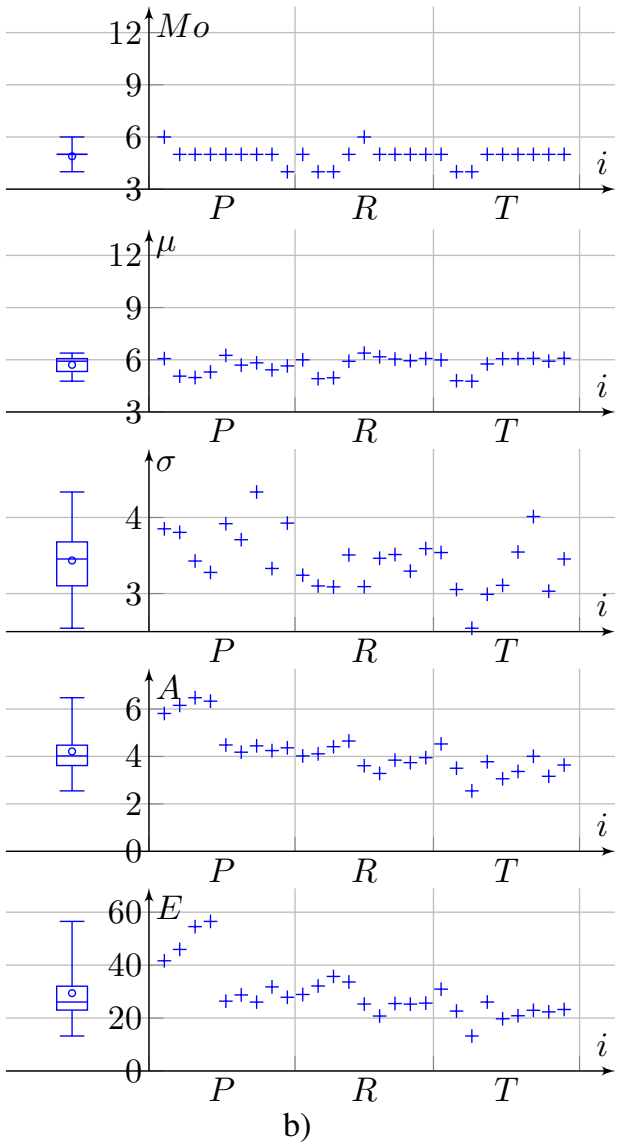


Рис. 2. Характеристики экспериментально полученных оценок РДП: a) включая подсеть провайдера корневого узла, b) исключая подсеть провайдера корневого узла

Для каждой пары α, β с помощью регулярных выражений определялось положение шлюза провайдера, при этом получались два значения длины пути: полная длина, включающая подсеть провайдера корневого узла

$$\tilde{\lambda}(\alpha, \beta) = k, \tag{7}$$

и скорректированная длина, исключающая подсеть провайдера корневого узла (то есть включающая только пересылки между узлами с белыми IP-адресами):

$$\lambda(\alpha, \beta) = k - w. \tag{8}$$

Для оконечных узлов β из подсети провайдера α принималось $\lambda(\alpha, \beta) = 1$.

После окончания измерений оба множества полученных значений длин путей $\tilde{\lambda}$ и λ обрабатывались GNU Octave. Соответственно, для каждого измерения i получались две оценки РДП:

– $\tilde{\zeta}_i(x)$, включающая подсеть провайдера корневого узла (рис. 1, a) и 2, a);
– $\zeta_i(x)$, исключающая подсеть провайдера корневого узла (рис. 1, b) и 2, b).

На рис. 2, a) и b) показаны соответственно характеристики оценок РДП $\tilde{\zeta}_i(x)$ и $\zeta_i(x)$ (мода Mo, среднее μ, среднеквадратическое отклонение σ, асимметрия A и эксцесс E). Значение эксцесса E вычисляется так, чтобы для нормального распределения он был бы равен нулю. Оси абсцисс соответствует идентификатор измерения i согласно (5). Таким образом, каждая точка справа от оси ординат соответствуют значению характеристики одной оценки РДП. Разброс этих значений иллюстрируют ящики с усами слева от оси ординат:

– окружность показывает усреднённое по всем измеренным оценкам РДП (среднее арифметическое) значение рассматриваемой характеристики;
– нижняя граница уса показывает минимальное значение характеристики;
– нижняя граница ящика показывает первую (нижнюю) квартиль;
– горизонтальная линия внутри ящика показывает медиану (вторую квартиль);
– верхняя граница ящика показывает третью (верхнюю) квартиль;
– верхняя граница уса показывает максимальное значение характеристики.

Анализ полученных результатов показал, что исключение из рассмотрения подсети провайдера корневого узла приводит к уменьшению разброса моды Mo и среднего μ и мало влияет на среднеквадратическое отклонение σ, асимметрию A и эксцесс E.

Кроме того, если изначально ожидалось, что определяющее влияние на РДП оказывает выбор оконечных узлов (соответственно, измерения на рис. 2 сгруппированы согласно (5) по множествам оконечных узлов), то рис. 1 и 2 показывают, что на форму и характеристики измеренного РДП влияет в основном корневой узел (даже после исключения подсети его провайдера).

Значения асимметрии и эксцесса как $\tilde{\zeta}_i(x)$, так и $\zeta_i(x)$ составляют (здесь и далее указано среднее значение без учёта выбросов, а в скобках — пятидесятипроцентный интервал):

$$\begin{cases} A_i \approx 3{,}5 \quad (3 \ldots 4{,}5) \\ E_i \approx 24 \quad (20 \ldots 30) \end{cases} \tag{9}$$

то есть обе эти характеристики существенно больше нуля. Ни для какого измерения i ни $\tilde{\zeta}_i(x)$, ни $\zeta_i(x)$ не имеют отрицательной асимметрии либо эксцесса.

Тем не менее, полученные по методу КУОУ оценки РДП могут отличаться от полного РДП глобальной сети. Так как точно измерить полное РДП глобальной сети не представляется возможным, для оценки корректности КУОУ необходимо рассмотреть модель сети, полное РДП которой доступно, и имитировать проведённый эксперимент для неё.

2. Имитационное моделирование погрешности эксперимента

Определение 5. *Будем называть количество вершин (узлов)* $|V_G|$ *графа* G размером графа.

Для проверки корректности полученных оценок РДП и его характеристик было произведено имитационное моделирование погрешности эксперимента по схеме КУОУ. Моделью глобальной сети был выбран граф G, вершины которого соответствуют узлам сети, а рёбра — возможности прямой пересылки данных на сетевом уровне. С учётом отсутствия петель на сетевом уровне G должен быть связным деревом.

Целью моделирования является сопоставление полного РДП $\zeta(x)$ и его оценок $\zeta_i(x)$, а также их характеристик, в частности, асимметрии и эксцесса.

Таким образом, размер G должен позволять:

– найти расстояния $\lambda(\alpha, \beta)$ для всех возможных пар вершин $\alpha, \beta \in V_G$ и рассчитать полное РДП $\zeta(x)$;
– выполнить имитацию схемы КУОУ, получить несколько оценок РДП $\zeta_i(x)$ и сравнить их с полным РДП.

Размеры множеств оконечных узлов, использованных в экспериментальном исследовании, составляют от 500 до 5000 узлов. Соответственно, минимальным размером, позволяющим выполнить оценивание РДП по схеме КУОУ, было принято $|V_G| = 10^3$ вершин [4]. Максимальным размером, при котором можно выполнить расчёт за приемлемое время, после распараллеливания оказался $|V_G| = 10^6$ вершин.

Граф G представлялся в памяти компьютера в виде сжатой матрицы смежности, то есть вектором ρ_G из $|V_G|$ списков, каждый из которых соответствует вершине графа и содержит номера смежных с ней вершин. Для расчёта длин путей использован поиск в ширину, позволяющий для заданной вершины $\alpha \in V_G$ рассчитать расстояния до всех прочих вершин графа G; его сложность для представления графа в виде ρ_G равна $O(|V_G| + |E_G|)$ [5]. При расчёте полного РДП поиск в ширину необходимо выполнить для всех вершин, таким образом, сложность возрастает в $|V_G|$ раз. Для дерева сложность поиска в ширину составляет $O(|V_G|)$, так что сложность расчёта полного РДП составляет $O(|V_G|^2)$.

Для каждого графа рассчитывалось несколько оценок РДП $\zeta_i(x)$ по схеме КУОУ. Для каждой оценки i корневая вершина $\alpha \in V_G$, соответствующая корневому узлу в эксперименте и оконечные вершины $Q \subset V_G$, соответствующие оконечным узлам, выбирались заново случайным образом. Количество оконечных вершин $|Q|$ во время предварительных вычислений варьировалось в диапазоне 500 … 5000. Так как не было выявлено существенного различия результатов, для итогового моделирования было принято $|Q| = 500$.

Рост графа и расчёт распределения длин путей реализованы на C++ (стандарт C++11). Для ускорения расчёт распараллелен с помощью OpenMP, что позволило вырастить дерево из $|V_G| = 10^6$ вершин, обработать его и сохранить результаты в текстовом файле менее чем за двое суток. Дальнейшие расчёты проводились в GNU Octave.

Одному прогону r модели роста γ для $|V_G| = 10^v$ соответствует одно дерево $G_r^{\gamma,v}$ и, соответственно, одно полное РДП $\zeta^{G_r^{\gamma,v}}(x)$, а также несколько его оценок $\zeta_i^{G_r^{\gamma,v}}(x)$ (здесь i — порядковый номер оценки). Таким образом, дереву $G_r^{\gamma,v}$ соответствует один набор характеристик полного РДП $\left(Mo(G_r^{\gamma,v}), \mu(G_r^{\gamma,v}), \sigma(G_r^{\gamma,v}), A(G_r^{\gamma,v}), E(G_r^{\gamma,v})\right)$ и множество наборов характеристик его оценок $\left(Mo_i(G_r^{\gamma,v}), \mu_i(G_r^{\gamma,v}), \sigma_i(G_r^{\gamma,v}), A_i(G_r^{\gamma,v}), E_i(G_r^{\gamma,v})\right)$. Для краткости будем писать $\zeta(x)$ и $\zeta_i(x)$, а также Mo, μ, σ, A, E и $Mo_i, \mu_i, \sigma_i, A_i, E_i$, если это не приведёт к неоднозначности.

2.1. Модель Барабаши—Альберт

Чаще всего для моделирования растущих сетей [3, 6] применяется безмасштабная модель Барабаши—Альберт [7] (далее будем обозначать эту модель γ = BA), полагающая, что узлы присоединяются к сети постепенно, следуя *принципу предпочтительного присоединения,* то есть вероятность p_j присоединения новой вершины v_{n+1} к каждой из существующих вершин $v_j, j \in \{1, \dots n\}$ пропорциональна количеству $|v_j|$ уже имеющихся связей:

$$p_j = \frac{|v_j|}{\sum_{k=1}^{n} |v_k|}. \tag{10}$$

Соответственно, первыми РДП и их оценки по схеме КУОУ были рассчитаны именно для безмасштабных деревьев $G_r^{\mathrm{BA},v}$ (рис. 3 и 4).

На рис. 3, а) показаны полные РДП для различных деревьев Барабаши—Альберт из 10^5 вершин ($G_r^{\mathrm{BA},5}$). Видно, что они мало отличаются друг от друга, всегда унимодальны и имеют схожую с нормальной вершину и тонкие хвосты.

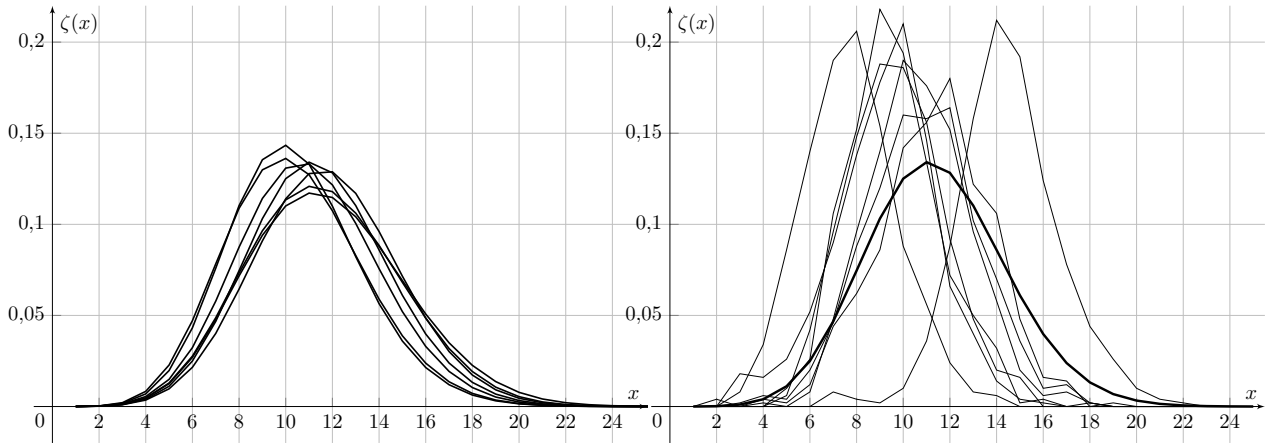


Fig. 3. Full NND distributions dispersion for different scale-free trees $G_r^{\mathrm{BA},5}$ (a); full NND distribution $\zeta(x)$ (thick line) and its estimates $\zeta_i(x)$ (thin lines) for the $G_4^{\mathrm{BA},5}$ (b)

Рис. 3. Разброс полных РДП для различных деревьев Барабаши—Альберт $G_r^{\mathrm{BA},5}$ (a); полное РДП $\zeta(x)$ (жирная линия) и его оценки $\zeta_i(x)$ (тонкие линии) для $G_4^{\mathrm{BA},5}$ (b)

На рис. 3, b) для одного из этих деревьев (полученного на четвёртом прогоне, то есть $G_4^{\mathrm{BA},5}$) показано полное РДП и его оценки. Большинство оценок $\zeta_i(x)$ рис. 3, b) выглядит более островершинными, чем полное $\zeta(x)$, но менее островершинными, чем экспериментально полученные оценки РДП глобальной сети (рис. 1, a, b).

Рассмотрим подробнее характеристики РДП. Так как общее количество полученных в результате моделирования оценок РДП для различных деревьев измеряется сотнями, характеристики каждой отдельной оценки РДП $\zeta_i(x)$ не могут быть приведены в рамках статьи. Соответственно, далее будем рассматривать различные деревья $G_r^{\mathrm{BA},v}$ и разброс характеристик оценок, полученных для каждого из них (для экспериментально полученных оценок РДП, полученных для глобальной сети, аналогичный разброс показывают ящики с усами в левой части рис. 2 a) и b).

На рис. 4 показаны характеристики полных РДП и их оценок для множества деревьев разного размера. Одна вертикаль соответствует одному прогону (одному дереву $G_r^{\mathrm{BA},v}$). Прогоны сгруппированы по размеру дерева $|V_G| = 10^v$ (показан вверху рисунка).

Внутри группы ось абсцисс соответствует номеру прогона модели r (показан вверху рисунка под $|V_G|$). Ниже показаны характеристики РДП дерева $G_r^{\mathrm{BA},v}$ и его оценок. Для каждой характеристики χ (где χ может быть модой Mo, средним μ, среднеквадратическим отклонением σ, асимметрией A и эксцессом E) и каждого дерева $G_r^{\mathrm{BA},v}$:

- чёрной точкой показано значение характеристики полного РДП $\chi(G_r^{\mathrm{BA},v})$;
- ящиком с усами на той же вертикали в тех же осях показан разброс характеристик оценок РДП $\chi_i(G_r^{\mathrm{BA},v})$ по i: среднее арифметическое по всем оценкам i (окружность), минимальное значение (нижняя граница уса), первая квартиль (нижняя граница ящика), медиана (линия внутри ящика), третья квартиль (верхняя граница ящика) и максимальное значение (верхняя граница уса).

Прежде всего из рис. 4 видно, что измерение РДП по схеме КУОУ в общем случае не позволяет получить полное РДП графа. Характеристики получаемых оценок значительно отличаются от характеристик полных РДП соответствующих деревьев.

Мода Mo_i и средняя длина пути μ_i для оценок РДП для деревьев небольших размеров (для $|V_G|$, сравнимых с объёмом выборки оконечных узлов) могут рассматриваться как оценки Mo и μ

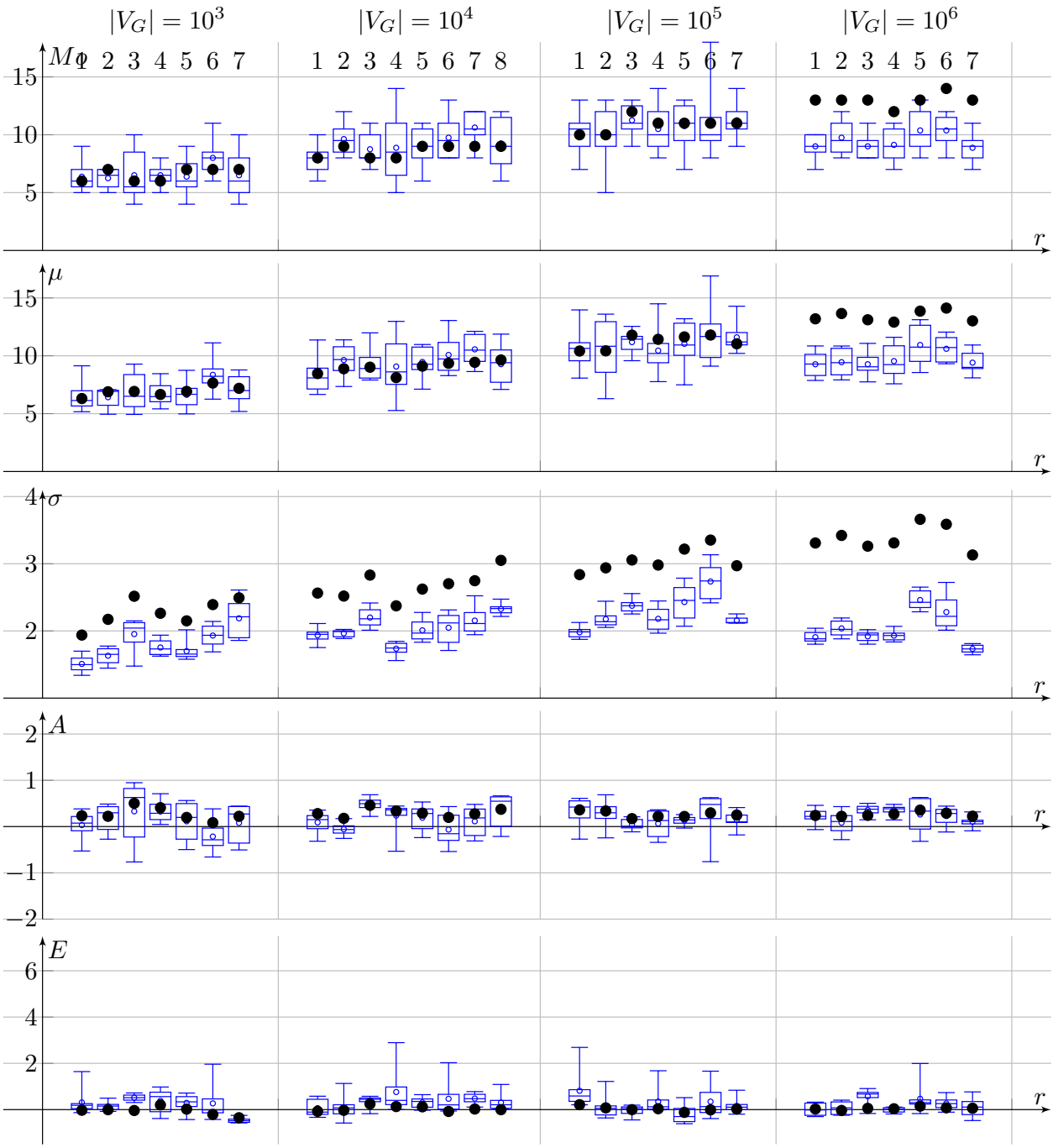


Fig. 4. Characteristics of full NND distribution in the scale-free trees $G_r^{\mathrm{BA},v}$ (black dots) and dispersion of the characteristics of the estimates (boxplots)

Рис. 4. Характеристики полного РДП в деревьях Барабаши—Альберт $G_r^{\mathrm{BA},v}$ (чёрные точки) и разброс характеристик его оценок (ящики с усами)

полного РДП, но уже для $|V_G| = 10^6$ значения Mo_i и μ_i существенно ниже Mo и μ соответственно. Таким образом, по полученным в результате эксперимента для глобальной сети по схеме КУОУ значениям Mo_i и μ_i также нельзя судить даже о порядке величины Mo и μ полного РДП глобальной сети.

Среднеквадратическое отклонение σ_i всех оценок РДП существенно ниже среднеквадратического отклонения σ полного РДП для всех деревьев, причём при увеличении количества узлов $|V_G| = 10^v$ растёт и систематическая погрешность оценивания.

Тем не менее рис. 4 показывает и то, что для каждого $|V_G|$ значения Mo_i, μ_i и σ_i оценок РДП имеют не вполне произвольные значения, а коррелируют со значениями Mo, μ и σ полного РДП.

Рассмотрим асимметрию и эксцесс полученных РДП. Асимметрия A полных РДП деревьев Барабаши—Альберт $\zeta(x)$ имеет небольшое положительное значение, а эксцесс E близок к нулю:

$$\begin{cases} A \approx 0{,}3 & (0{,}2 \dots 0{,}4) \\ E \approx 0 & (-0{,}1 \dots +0{,}2) \end{cases}, \tag{11}$$

С ростом количества $|V_G|$ вершин в дереве эти величины стабилизируются.

Отличие асимметрии A_i и эксцесса E_i оценок РДП от значений A и E полных РДП составляет сотни процентов, что связано прежде всего с близостью A и E к нулю. При этом асимметрия при оценивании по схеме КУОУ немного занижается, а эксцесс — немного завышается:

$$\begin{cases} A_i \approx 0{,}2 & (0{,}0 \dots 0{,}3) \\ E_i \approx 0{,}3 & (-0{,}2 \dots +0{,}8) \end{cases}. \tag{12}$$

Таким образом:

- с ростом количества $|V_G|$ вершин в дереве не растут ни асимметрия A и эксцесс E полных РДП деревьев Барабаши—Альберт, ни их оценки по КУОУ A_i и E_i;
- хотя бы одна оценка E_i для большинства прогонов отрицательна;
- ни одна оценка A_i и E_i ни для какого прогона не достигает полученных в результате эксперимента значений (9) и даже не приближается к ним.

Таким образом, различие между асимметрией и эксцессом экспериментально полученных для сетевого уровня глобальной сети данных и распределениями длин путей в деревьях Барабаши—Альберт обусловлено не случайными отклонениями, не методикой измерения и не отличием в размерах.

Соответственно, различие обусловлено выбранной моделью роста дерева, то есть модель Барабаши—Альберт не отражает свойств глобальной сети [8].

2.2. Модифицированные модели

Отличие экспериментально полученных оценок A_i и E_i от предсказываемых безмасштабной моделью Барабаши—Альберт может быть обусловлено различными аспектами роста реальной глобальной сети. Тем не менее, сама концепция постепенного роста сети и предпочтительного присоединения логична, так что была предпринята попытка внести в модель Барабаши—Альберт небольшие изменения и исследовать их влияние на асимметрию и эксцесс РДП выращиваемых графов.

Были выдвинуты три гипотезы о том, какой именно фактор оказывает определяющее влияние на значения асимметрии и эксцесса оценок РДП:

1. Ограничение степени узла.

   В оригинальной модели Барабаши—Альберт предполагается, что количество связей $|\nu_j|$ вершины может быть сколь угодно большим. Это верно для связей, не создающих нагрузку на узлы (таких, как web-ссылки), но неверно для сетевого уровня. Предположим, что количество связей вершины ограничено сверху некоторым числом M:

   $$|\nu_j| \leqslant M \tag{13}$$

   Таким образом, новая вершина не может соединяться ребром с вершиной, уже имеющей M связей; среди остальных вершин выбор происходит по принципу предпочтительного присоединения.

2. Удаление узлов.
   Узлы не только добавляются к глобальной сети, но и отключаются от неё. Рассмотрим модель, в которой на каждом шаге либо (с некоторой вероятностью p) удаляется случайно выбранная вершина, либо добавляется новая (с вероятностью $1 - p$). При удалении вершины её связи перераспределяются между оставшимися, чтобы сохранить связность.
3. Квазипредфрактальная структура сети.
   Глобальная сеть не гомогенна. Она состоит из множества подсетей различных провайдеров, причём каждый провайдер, предоставляющий связь конечным пользователям, сам пользуется услугами более крупного (магистрального) провайдера.
   Опишем рост графа G как совокупности нескольких подграфов G_s:

$$G = \bigcup_{s=1}^{m} G_s \tag{14}$$

На каждом шаге либо к одному из G_s присоединяется новая вершина, либо создаётся новый подграф G_{m+1}. Вероятности этих событий пропорциональны размерам G_s и G_{m+1}, то есть также соответствуют принципу предпочтительного присоединения. Новая вершина к подграфу G_s присоединяется аналогично модели Барабаши—Альберт. При создании нового подграфа G_{m+1} для него также по принципу предпочтительного присоединения выбирается магистральный провайдер — один из существующих подграфов G_s, далее G_{m+1} соединяется с G_s.
Таким образом, соединение подграфов G_s в графе G аналогично соединению вершин в каждом из подграфов. При точном совпадении структуры граф G можно было бы назвать предфрактальным [9].

Определение 6. *Будем называть* квазипредфрактальным *граф G, состоящий из нескольких подграфов G_s, соединённых рёбрами, причём модель роста всех G_s одинакова и аналогична модели роста самого G.*

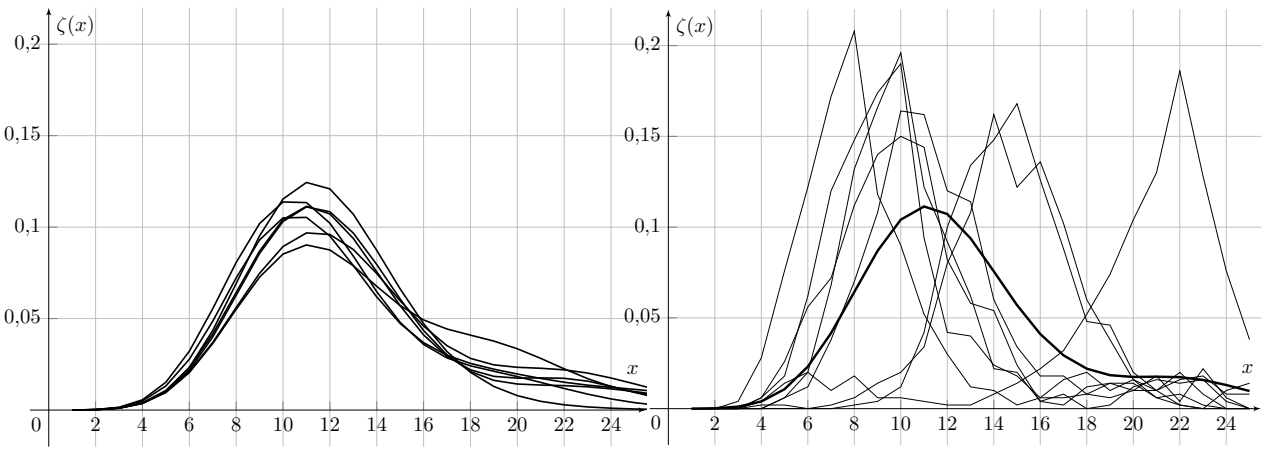


Fig. 5. Full NND distributions scattering in the quasi-pre-fractal trees $G_r^{\mathrm{PF},5}$ (a); full NND distribution $\zeta(x)$ (thick line) and its estimates $\zeta_i(x)$ (thin lines) for the $G_1^{\mathrm{PF},5}$ (b)

Рис. 5. Разброс полных РДП для квазипредфрактальных деревьев $G_r^{\mathrm{PF},5}$ (a); полное РДП $\zeta(x)$ (жирная линия) и его оценки $\zeta_i(x)$ (тонкие линии) для $G_1^{\mathrm{PF},5}$ (b)

Соответственно, в исследовании были рассмотрены три модели роста, каждая из которых реализует только одну из описанных гипотез. Модели, реализующие ограничение степени узла и удаление узлов из сети, приводят к результатам, качественно не отличающимся от модели Барабаши—Альберт, и, соответственно, не рассматриваются в данной статье.

Квазипредфрактальная модель, описывающая рост сети как рост множества связанных друг с другом подсетей (будем обозначать её γ = PF), приводит к отличным от Барабаши—Альберт результатам (рис. 5 и 6).

Полные РДП $\zeta(x)$ на рис. 5 отличаются от рис. 3 наличием более или менее толстого правого хвоста. Его толщина обусловлена соотношением размеров подграфов G_s и порядком их соединения.

Полное РДП квазипредфрактального дерева может иметь:

- более или менее толстый правый хвост, аналогичный рис. 5 (с крупным подграфом соседствует несколько меньших — наиболее распространённый среди приведённых результатов вариант);
- один или несколько ярко выраженных побочных максимумов (два или три конкурирующие подграфа сопоставимых размеров — $G_2^{\mathrm{PF},4}$, рис. 7, a);
- тонкий хвост, как для РДП деревьев Барабаши—Альберт (сформировался только один крупный подграф; прочие, если и существуют, ничтожно малы — $G_1^{\mathrm{PF},6}$, рис. 7, b).

В последнем случае, когда граф $G_r^{\mathrm{PF},v}$ фактически совпадает со своим крупнейшим подграфом, его можно считать выращенным по модели Барабаши—Альберт. Соответственно, и асимметрия и эксцесс полного РДП, и их оценки будут близки к нулю и иметь малый разброс. Действительно,

$$\begin{cases} A_i(G_1^{\mathrm{PF},6}) \approx A(G_1^{\mathrm{PF},6}) \approx 0{,}3 \\ E_i(G_1^{\mathrm{PF},6}) \approx E(G_1^{\mathrm{PF},6}) \approx 0 \end{cases}. \tag{15}$$

В случае нескольких максимумов и, соответственно, нескольких крупных подграфов, асимметрия и эксцесс полного РДП могут быть как положительными, так и отрицательными, в зависимости от взаимного расположения максимумов. Так, $E(G_2^{\mathrm{PF},4}) < 0$.

Так как подграфы $G_s \subset G_r^{\mathrm{PF},v}$ и связи между ними организуются случайно в процессе роста графа $G_r^{\mathrm{PF},v}$, форма полных РДП для разных прогонов r различается сильнее, чем для модели Барабаши—Альберт. Соответственно, все характеристики на рис. 6 (организованном аналогично рис. 4) имеют большой разброс, не уменьшающийся с увеличением размеров деревьев $|V_G|$.

Максимально возможные значения асимметрии и эксцесса полных РДП квазипредфрактальных деревьев растут с увеличением $|V_G|$. Так, для квазипредфрактальных деревьев из $10^5 \ldots 10^6$ вершин усреднённые асимметрия и эксцесс без учёта выбросов:

$$\begin{cases} A \approx 1 \quad (0{,}5 \ldots 1{,}2) \\ E \approx 1{,}2 \quad (0{,}5 \ldots 1{,}8) \end{cases}, \tag{16}$$

и растут с ростом $|V_G|$ (при этом увеличивается и разброс распределений длин путей).

Из рис. 6 видно, что характеристики оценок РДП по схеме КУОУ для квазипредфрактальных деревьев отличаются от характеристик полных РДП ещё больше, чем для деревьев Барабаши—Альберт, что подтверждает сделанный ранее вывод о том, что результаты, полученные по КУОУ экспериментально, не могут рассматриваться как характеристики РДП глобальной сети.

При оценивании по схеме КУОУ асимметрия и эксцесс унимодальных полных РДП с толстым хвостом чаще завышаются, чем занижаются, причём большие значения A и E завышаются сильнее. Так, для $|V_G| \in [10^5, 10^6]$ вершин:

$$\begin{cases} A_i \approx 1{,}2 \quad (1{,}0 \ldots 1{,}8) \\ E_i \approx 2{,}7 \quad (1{,}7 \ldots 4{,}3) \end{cases}, \tag{17}$$

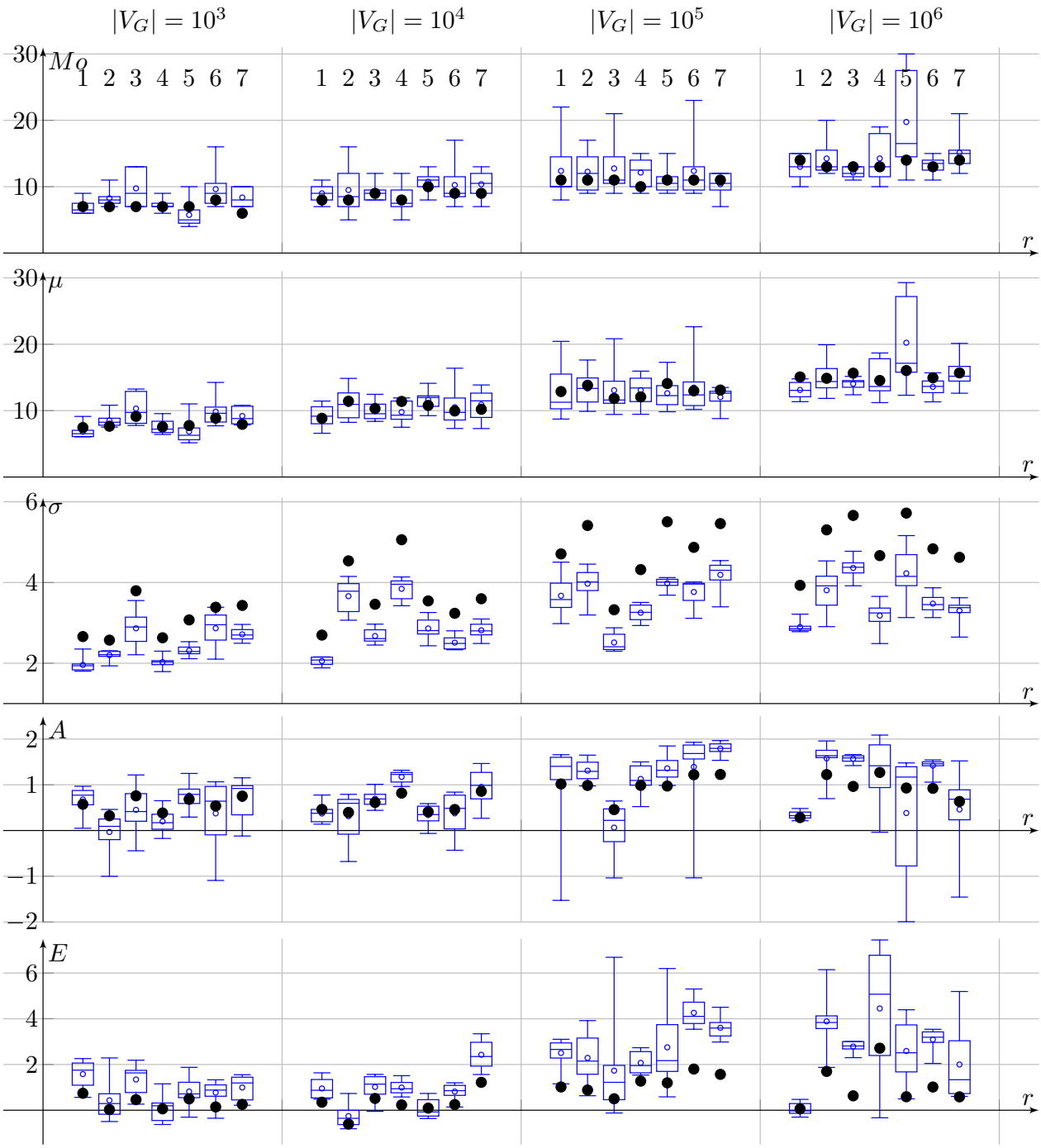


Fig. 6. Characteristics of full NND distribution in the quasi-pre-fractal trees $G_r^{\mathrm{PF},v}$ (black dots) and scattering of characteristics of estimates (boxplots)

Рис. 6. Характеристики полного РДП в квазипредфрактальных деревьях $G_r^{\mathrm{PF},v}$ (чёрные точки) и разброс характеристик его оценок (ящики с усами)

и эти величины растут с ростом $|V_G|$. Также необходимо отметить, что для большинства прогонов $E_i > A_i$ (для асимметрии A и эксцесса E полных РДП это выполняется реже).

Таким образом, для $|V_G| \in [10^8, 10^{10}]$ вершин оценки асимметрии и эксцесса для некоторых квазипредфрактальных деревьев будут приближаться к экспериментально полученным значениям (9).

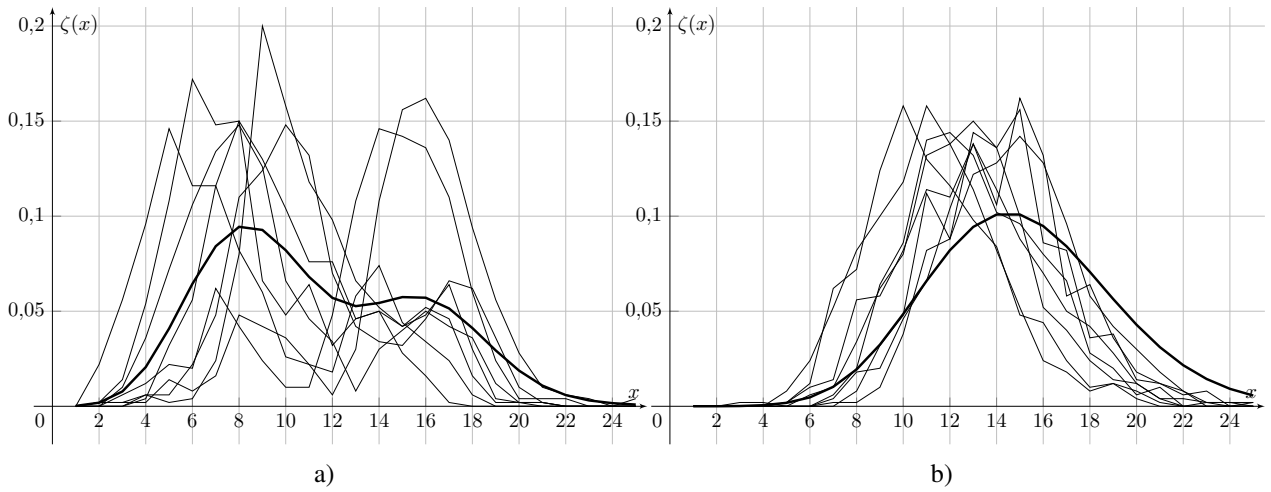


Fig. 7. Full NND distribution $\zeta(x)$ (thick line) and its estimates $\zeta_i(x)$ (thin lines) for: a) $G_2^{\mathrm{PF},4}$, b) $G_1^{\mathrm{PF},6}$

Рис. 7. Полное РДП $\zeta(x)$ (жирная линия) и его оценки $\zeta_i(x)$ (тонкие линии): a) для $G_2^{\mathrm{PF},4}$, b) для $G_1^{\mathrm{PF},6}$

Заключение

Для оценивания РДП сетевого уровня глобальной сети была разработана схема КУОУ, использующая инструментальные средства, стандартные для большинства дистрибутивов GNU/Linux, и проведена серия измерений, начатых в 2011 году; результаты измерений были обработаны с помощью GNU Octave. Полученные оценки РДП имеют высокие положительные значения асимметрии и эксцесса.

Чтобы установить возможную ошибку, вносимую таким оцениванием, было проведено имитационное моделирование измерений. Его результаты показали, что погрешность оценок характеристик РДП по КУОУ может составлять 100% и выше, так что полученные в результате измерений данные не могут рассматриваться как характеристики РДП сетевого уровня глобальной сети или их адекватные оценки.

Тем не менее, также имитационное моделирование показало, если рост сети описывается безмасштабной моделью Барабаши—Альберт, то при оценивании РДП по схеме КУОУ не могут быть получены настолько высокие значения асимметрии и эксцесса, как у полученных экспериментально оценок РДП. Таким образом, проведённые измерения в сочетании с имитационным моделированием позволяют уверенно утверждать, что сетевой уровень глобальной сети не описывается безмасштабной моделью Барабаши—Альберт.

Разработанная квазипредфрактальная модель роста в некотором приближении может быть использована для описания сетевого уровня глобальной сети. Имитационное моделирование показывает, что в квазипредфрактальном дереве, состоящем из одной крупнейшей подсети и множества более мелких, размер которого сопоставим с глобальной сетью, при оценивании РДП по КУОУ будут получены высокие значения асимметрии и эксцесса, сопоставимые с полученными экспериментально.

References

[1] A. Gorodilov, A. Kononova, and V. Shangin, "Osobennosti peredachi dannyh v decentralizovannyh piringovyh setyah", *Proceedings of Universities. Electronics*, vol. 98, no. 6, pp. 95–97, 2012.

[2] A. P. Shiryaev, A. V. Dorofeev, A. R. Fedorov, L. G. Gagarina, and V. V. Zaycev, "LDA models for finding trends in technical knowledge domain", in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, 2017, pp. 551–554.

[3] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[4] A. Fronczak, P. Fronczak, and J. A. Hołyst, "Average path length in random networks", *Physical Review E*, vol. 70, no. 5, p. 056 110, 2004.

[5] R. Sedgewick, *Algorithms in C, Part 5: Graph Algorithms*, 2002.

[6] H. Jeong, B. Tombor, Z. N. Albert R.and Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks", *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[7] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks", *Reviews of modern physics*, vol. 74, no. 1, pp. 47–97, 2002.

[8] A. Gorodilov A.V.and Kononova, "Simulation as tool of error assessment of experimental measurement of path-lengths distribution in global network", *Sistemy komp'yuternoj matematiki i ih prilozheniya: materialy XX Mezhdunarodnoj nauchnoj konferencii*, vol. 1, pp. 34–41, 2019.

[9] R. A. Kochkarov, D. A. Pavlov, and D. A.-Z. Hubieva, "Fractal and preefactal graphs, basic definitions and symbols", *Scientific journal of KubSAU (Polythematic online scientific journal of Kuban State Agrarian University)*, no. 134, pp. 174–188, 2017.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

THEORY OF DATA

On the Properties of Algebraic Geometric Codes as Copy Protection Codes

V. M. Deundyak[1;2], D. V. Zagumennov[1]

DOI: 10.18255/1818-1015-2020-1-22-38

[1]Southern Federal University, 105/42 Bolshaya Sadovaya str., Rostov-on-Don 344006, Russia.
[2]FGNU NII Specvusavtomatika, 51 Gazetniy lane, Rostov-on-Don 344002, Russia.

MSC2020: 94B27
Research article
Full text in Russian

Received November 19, 2019
After revision February 17, 2020
Accepted February 28, 2020

Traceability schemes which are applied to the broadcast encryption can prevent unauthorized parties from accessing the distributed data. In a traceability scheme a distributor broadcasts the encrypted data and gives each authorized user unique key and identifying word from selected error-correcting code for decrypting. The following attack is possible in these schemes: groups of c malicious users are joining into coalitions and gaining illegal access to the data by combining their keys and identifying codewords to obtain pirate key and codeword. To prevent this attacks, classes of error-correcting codes with special c-FP and c-TA properties are used. In particular, c -FP codes are codes that make direct compromise of scrupulous users impossible and c -TA codes are codes that make it possible to identify one of the attackers. We are considering the problem of evaluating the lower and the upper boundaries on c, within which the L-construction algebraic geometric codes have the corresponding properties. In the case of codes on an arbitrary curve the lower bound for the c-TA property was obtained earlier; in this paper, the lower bound for the c-FP property was constructed. In the case of curves with one infinite point, the upper bounds for the value of c are obtained for both c-FP and c-TA properties. During our work, we have proved an auxiliary lemma and the proof contains an explicit way to build a coalition and a pirate identifying vector. Methods and principles presented in the lemma can be important for analyzing broadcast encryption schemes robustness. Also, the c-FP and c-TA boundaries monotonicity by subcodes are proved.

Keywords: error-correcting codes; traceability schemes; algebraic geometry codes

INFORMATION ABOUT THE AUTHORS

Vladimir M. Deundyak correspondence author	orcid.org/0000-0001-8258-2419. E-mail: vl.deundyak@gmail.com PhD.
Denis V. Zagumennov	orcid.org/0000-0001-8990-9058. E-mail: zagumionnov.denis@yandex.ru graduate student.

For citation: V. M. Deundyak and D. V. Zagumennov, "On the Properties of Algebraic Geometric Codes as Copy Protection Codes", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 22-38, 2020.

© Deundyak V. M., Zagumennov D. V., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Исследование свойств АГ-кодов как кодов для защиты от копирования

В. М. Деундяк[1;2], Д. В. Загуменнов[1]

DOI: 10.18255/1818-1015-2020-1-22-38

[1]Южный Федеральный Университет, ул. Большая Садовая, 105/42, Ростов-на-Дону, 344006, Россия.
[2]ФГНУ НИИ Спецвузавтоматика, пер. Газетный, 51, Ростов-на-Дону, 344002, Россия.

УДК 519.7
Научная статья
Полный текст на русском языке

Получена 19 ноября 2019 г.
После доработки 17 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

Схемы специального широковещательного шифрования используются для защиты легально тиражируемой цифровой продукции от несанкционированного копирования. В таких схемах распространитель тиражирует данные свободно в зашифрованном виде, а для расшифрования выдаёт каждому легальному пользователю уникальный набор ключей и идентифицирующих векторов из некоторого помехоустойчивого кода. Однако, в этих схемах возможна атака, в ходе которой группы из c недобросовестных пользователей могут объединяться в коалиции и получать нелегальный доступ к данным, комбинируя выданную им ключевую информацию для получения пиратской ключевой информации — идентификационного вектора и ключа. Для борьбы с коалиционными атаками применяются классы помехоустойчивых кодов, обладающих специальными c-FP и c-TA свойствами. Класс c-FP-кодов составляют коды, исключающие возможность прямой компрометации добросовестных пользователей, а класс c-TA-кодов составляют коды, позволяющие гарантированно определить одного из злоумышленников. Рассматривается задача нахождения нижних и верхних границ значения величины c, в пределах которых алгеброгеометрические коды L-конструкции обладают соответствующими свойствами. В случае кодов на произвольной кривой ранее была получена нижняя граница для свойства c-TA, в настоящей работе построена нижняя граница для свойства c-FP. В случае кривых с одной бесконечной точкой получены верхние границы значения c как для c-FP, так и для c-TA свойств. При нахождении этих границ получена вспомогательная конструктивная лемма, в доказательстве которой содержится явный способ построения коалиции и пиратского идентификационного вектора; этот способ важен при анализе стойкости схем широковещательного шифрования. Доказаны свойства монотонности рубежей c-FP и c-TA свойств по подкодам.

Ключевые слова: помехоустойчивое кодирование; широковещательное шифрование; алгеброгеометрические коды

ИНФОРМАЦИЯ ОБ АВТОРАХ

Владимир Михайлович Деундяк
автор для корреспонденции
orcid.org/0000-0001-8258-2419. E-mail: vl.deundyak@gmail.com
канд. физ.-мат. наук, доцент.

Денис Владимирович Загуменнов
orcid.org/0000-0001-8990-9058. E-mail: zagumionnov.denis@yandex.ru
аспирант.

Для цитирования: V. M. Deundyak and D. V. Zagumennov, “On the Properties of Algebraic Geometric Codes as Copy Protection Codes”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 22-38, 2020.

© Деундяк В. М., Загуменнов Д. В., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

В работе рассматривается перспективный способ применения помехоустойчивых АГ-кодов L-конструкции в качестве кодов для защиты легально тиражируемой цифровой продукции от несанкционированного копирования [1], который называется схемой специального широковещательного шифрования (ССШШ). В этих схемах распространитель тиражирует данные свободно в зашифрованном виде, а каждому легальному пользователю для расшифрования данных выдаёт уникальный набор ключей и идентифицирующих векторов из соответствующего помехоустойчивого кода. В ССШШ пользователи применяют кодовые идентифицирующие векторы при выполнении легального доступа к данным. В случае обнаружения нелегального использования ключевой информации её владелец может быть идентифицирован контролёром. В ССШШ возможны атаки следующего вида: некоторые недобросовестные легальные пользователи могут объединяться в коалиции злоумышленников некоторой мощности $c \in \mathbb{N} \setminus \{1\}$ с целью создания пиратских идентифицирующих векторов и ключей, которые можно использовать для выполнения нелегального доступа к данным, что может привести к различным злоупотреблениям. Для борьбы с подобными атаками в [1—3] предложен метод обнаружения членов коалиций, основанный на использовании некоторых классов линейных кодов, описание и анализ эффективности подобных схем приведён также в [4].

Для использования в таких системах в настоящее время активно исследуются и применяются классы так называемых c-TA и c-FP-кодов для защиты от несанкционированного копирования. Класс c-TA-кодов составляют такие коды, для которых применение к пиратскому идентификационному вектору любого декодера, работающего по минимуму кодового расстояния, позволяет гарантированно найти идентификационный вектор злоумышленника, входящего в атакующую коалицию мощности c. Более широкий класс c-FP-кодов составляют такие коды, для которых пиратский идентификационный вектор, созданный коалицией мощности c, не может являться идентификационным вектором пользователя, не входящего в коалицию, что исключает возможность прямой компрометации невиновных пользователей.

Актуальными представляются задачи поиска новых классов помехоустойчивых кодов для дальнейшего их использования в ССШШ, а также уточнения рубежей, при которых выполнены свойства c-TA и c-FP. В [3] показана возможность применения некоторых кодов Рида-Соломона в качестве c-TA-кодов, а в [5] исследованы рубежи значения c для кодов Рида-Соломона, при которых они являются c-TA и c-FP-кодами. В работе [6] показана возможность применения q-ичных кодов Рида-Малера в качестве как c-TA, так и c-FP-кодов, а также исследованы соответствующие рубежи. В [3] показана возможность применения некоторых алгеброгеометрических кодов (АГ-кодов) L-конструкции, а в [7] получены достаточные условия наличия свойства c-TA у АГ-кодов, а также условия применимости в ССШШ некоторых списочных декодеров для АГ-кодов L-конструкции.

В настоящей статье вычислена нижняя граница для рубежа c-FP свойства в случае АГ-кодов на произвольных кривых и доказана монотонность рубежей c-FP свойства и c-TA свойства по подкодам. Для АГ-кодов на специальных классах кривых с одной бесконечной точкой вычислены верхние границы рубежей как для c-FP, так и для c-TA свойства.

1. Классы c-TA и c-FP-кодов

Ниже будем использовать стандартные обозначения из теории кодирования (см. [8]). Пусть C – линейный $[n, k, d]_q$ код, $x, y \in C$,

$$I(x, y) = \{i \in \mathbb{N} \,:\, 1 \le i \le n, x_i = y_i\};$$

ясно, что $|I(x, y)| = n - d(x, y)$, где $d(x, y)$ – расстояние Хэмминга между x и y.

Пусть $c \in \mathbb{N} \setminus \{1\}$. Коалицией кода C назовём множество $C_0 = \{u_1, u_2, \dots, u_c\}$, где $u_i \in C$. Число cбудем называть мощностью коалиции, а множество коалиций кода C мощности не больше c будем

обозначать как $\mathrm{coal}_c(C)$. Ясно, что c существенно меньше мощности кода C. Множеством потомков коалиции C_0 назовём множество

$$\mathrm{desc}(\mathrm{C}_0) = \{(y_1, y_2, , \dots, y_n) \in \mathbb{F}_q^n \mid y_j = u_{i,j},\ i \in \{1, \dots, c\},\ j \in \{1, \dots, n\}\}.$$

Линейный код C называется c-ТА кодом ([1], определение 1.1), если выполняется следующее условие:

$$\forall\ C_0 \in \mathrm{coal}_c(C)\ \ \forall\ v \in (C \setminus C_0)\ \ \forall\ y \in \mathrm{desc}(\mathrm{C}_0)\ \ \exists\ \omega \in C_0\ d(w, y) < d(v, y).$$

Отметим, что если для кода C выполнено c-ТА-свойство, то для любого вектора $v \in C$ ни одна коалиция мощности не более c не сможет комбинированием элементов своих кодовых векторов сгенерировать потомка ω, находящегося ближе к v, чем к этой коалиции. Множеством ТА-компрометации для кода C называется множество:

$$\Omega_{TA}(C) = \{c \in \mathbb{N}_1 \ :\ \exists v \in C\ \exists C_0 \in \mathrm{coal}_c(C \setminus \{v\})\ \exists \omega \in \mathrm{desc}(C_0) \setminus C_0\ \forall u \in C_0\ :\ d(v, \omega) \le d(u, \omega)\}$$

(см. [6], с. 101). Таким образом, для того, чтобы доказать, что для кода C не выполнено c-ТА-свойство, достаточно построить кодовый вектор v, коалицию C_0 мощности максимум c и потомка этой коалиции ω такие, чтобы расстояние от этого потомка ω до v было меньше, чем расстояние от этого потомка ω до любого из членов коалиции.

Линейный код C будем называть c-FP кодом ([1], определение 1.1), если выполняется следующее условие:

$$\forall\ C_0 \in \mathrm{coal}_c(C)\ \ \forall\ z \in (C \setminus C_0)\ :\ z \notin \mathrm{desc}(C_0) \setminus C_0.$$

Таким образом, если для кода C выполнено c-FP-свойство, то ни одна коалиция мощности не более c не сможет комбинированием элементов своих кодовых векторов сгенерировать другой кодовый вектор. Множеством FP-компрометации для кода C называется множество:

$$\Omega_{FP}(C) = \{c \in \mathbb{N}_1 \ :\ \exists C_0 \in \mathrm{coal}_c(C)\ \exists z \in (C \setminus C_0)\ :\ z \in \mathrm{desc}(C_0) \setminus C_0\ (z \in \mathrm{desc}(C_0))\}$$

(см. [6], с. 101). Таким образом, для того, чтобы доказать, что для кода C не выполнено c-FP-свойство, достаточно построить коалицию мощности не более c и кодовый вектор такие, чтобы этот кодовый вектор являлся потомком этой коалиции.

Множества $\Omega_{TA}(C)$ и $\Omega_{FP}(C)$ являются целочисленными отрезками:

$$\Omega_{TA}(C) = \{R_{TA}(C), R_{TA}(C) + 1, \dots\},$$

$$\Omega_{FP}(C) = \{R_{FP}(C), R_{FP}(C) + 1, \dots\}.$$

Величины $R_{TA}(C)$ и $R_{FP}(C)$ будем называть рубежами множеств компрометации. Из определений вытекают следующие вложение и неравенство:

$$\Omega_{FP}(C) \subseteq \Omega_{TA}(C),\ R_{TA}(C) \le R_{FP}(C) \tag{1}$$

Докажем лемму о монотонности свойств ТА и FP.

Лемма 1. *Пусть C_1 и C_2 – линейные коды в $\mathbb{F}_q^n$, и C_1 – подкод C_2. Тогда выполняется:*

$$R_{TA}(C_1) \ge R_{TA}(C_2),\ \ R_{FP}(C_1) \ge R_{FP}(C_2).$$

Доказательство. Пусть $c \in \mathbb{N} \setminus \{1\}$, такое, что:

$$\exists v_1 \in C_1 \, \exists C^1 \in \mathrm{coal}_c(C_1 \setminus \{v_1\}) \, \exists \omega_1 \in \mathrm{desc}(C^1) \setminus C^1 \, \forall u \in C^1 \, : \, d(v_1, \omega_1) \leq d(u, \omega_1).$$

Тогда, учитывая, что $C_1 \subset C_2$, получаем, что

$$\exists v_2 \in C_2 = v_1 \, \exists C^2 \in \mathrm{coal}_c(C_2 \setminus \{v_2\}) = C^1 \, \exists \omega_2 \in \mathrm{desc}(C^2) \setminus C^2 = \omega_1 \, \forall u \in C^2 \, : \, d(v_2, \omega_2) \leq d(u, \omega_2).$$

Таким образом, если для $c \in \mathbb{N} \setminus \{1\}$ не выполняется c-ТА свойства для кода C_1, то c-ТА свойство не выполняется и для кода C_2, значит, $R_{TA}(C_1) \geq R_{TA}(C_2)$.

Аналогично доказывается и второе неравенство. □

Для кодов Рида-Маллера аналогичная лемма доказана в [9] (теоремы 2 и 4).

2. Алгеброгеометрические коды L-конструкции

2.1. Основные понятия

Ниже будем использовать подходы к АГ-кодам L-конструкции из [10, 11]. Рассмотрим конечное поле $\mathbb{F}_q$ и кольца многочленов $\mathbb{F}_q[x_1, x_2]$, $\mathbb{F}_q[X_1, X_2, X_3]$. Обозначим через $\mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$ множество однородных многочленов из $\mathbb{F}_q[X_1, X_2, X_3]$.

Отметим, что между многочленами f из $\mathbb{F}_q[x_1, x_2]$ и однородными многочленами F из $\mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$ существует взаимно-однозначное соответствие ([11], стр. 106-107), определяемое по следующему правилу. Если d – максимальная степень одночлена в многочлене $f \in \mathbb{F}_q[x_1, x_2]$, то F получается из f заменой каждого одночлена вида $x_1^i x_2^j$ на одночлен вида $X_1^i X_2^j X_3^{d-i-j}$. Это соответствие называется проективизацией.

Если точка P имеет аффинные координаты (a_1, a_2), то проективные координаты этой точки будем записывать так $(a_1 : a_2 : 1)$, в случае бесконечной точки третья проективная координата равна нулю ([10], с.7-8). Пусть $F \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$, $\mathcal{X} = \mathcal{X}(F, \mathbb{F}_q)$ – плоская гладкая проективная кривая над полем $\mathbb{F}_q$, заданная неприводимым многочленом $F \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$([11], п. 2.1.2). Далее в тексте будем рассматривать только такие кривые. Кривые имеют параметр $g \in \mathbb{N} \cup \{0\}$, называемый родом. В случае плоских гладких кривых род вычисляется по известной формуле ([11], следствие 2.2.8):

$$g = \frac{(\deg(F) - 1)(\deg(F) - 2)}{2}.$$

Через (F) обозначим главный идеал в $\mathbb{F}_q[X_1, X_2, X_3]$, порождённый F. В кольце

$$R = \left\{ \frac{P}{Q} \, : \, P, Q \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3], \deg(P) = \deg(Q), Q \notin (F) \right\}$$

с естественными операциями сложения и умножения рассмотрим максимальный идеал

$$I = \left\{ \frac{P}{Q} \, : \, P, Q \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3], \deg(P) = \deg(Q), Q \notin (F), P \in (F) \right\}$$

([11], п. 2.5.4). Тогда фактор-кольцо R/I является полем, оно называется полем рациональных функций на кривой $\mathcal{X}$ и обозначается $\mathbb{F}_q(\mathcal{X})$.

Согласно [11] (п. 2.5.2):

$$\forall M \in \mathcal{X} \, \exists T \in \mathbb{F}_q(\mathcal{X}) \, \forall H \in \mathbb{F}_q(\mathcal{X}) \exists U \in \mathbb{F}_q(\mathcal{X}) \, : \, T(M) = 0, U(M) \neq 0 \, : \, H = T^m U,$$

где $m \in \mathbb{Z}$, и значение величины m не зависит от выбора элемента T. Порядком $H = T^m U \in \mathbb{F}_q(\mathcal{X})$ в точке $M \in \mathcal{X}$ назовём значение величины m и будем обозначать это так: $\mathrm{ord}_M(H) = m$.

Дивизором D на проективной кривой $\mathcal{X}$ называется формальная сумма следующего вида: $D = \sum_{M\in\mathcal{X}} a_M M, a_M \in \mathbb{Z}$. Носителем дивизора называют множество $\mathrm{supp}(D) = \{M \in \mathcal{X} : a_M \neq 0\}$, а степенью дивизора D – число $\deg(D) = \sum a_M$. Если $\deg(D) = \alpha$, то иногда будем вместо D писать D_α. Говорят, что дивизор

$$D = \sum a_M M, a_M \in \mathbb{Z}, M \in \mathcal{X}$$

эффективен, если все $a_M \geq 0$. Этот факт обозначается следующим образом: $D \geq 0$. Дивизором функции $H \in \mathbb{F}_q(X)$ на проективной кривой $\mathcal{X}$ называется дивизор:

$$(H) = \sum_{M\in\mathcal{X}} \mathrm{ord}_M(H)M.$$

Для каждого $M \in \mathcal{X}$ рассмотрим произвольный линейный однородный многочлен $L \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$, для которого $L(M) \neq 0$, и многочлен $G \in \mathbb{F}_q^{\mathrm{hom}}[X_1, X_2, X_3]$, у которого $\deg(G) = r$. Пусть $I(M; \mathcal{X}; G) = \mathrm{ord}_M(G/L^r)$ (см. [10], определение 2.22). Дивизором пересечения G и $\mathcal{X}$ называется дивизор вида

$$\mathcal{X} \cdot G = \sum_{M\in\mathcal{X}} I(M; \mathcal{X}; G)M. \tag{2}$$

Зафиксируем дивизор $D = \sum_{M\in\mathcal{X}} a_M M$. Множество

$$L(D) = \{H \in \mathbb{F}_q(\mathcal{X}) : (H) + D \geq 0\}$$

называется пространством Римана-Роха, ассоциированным с D. Пространство $L(D)$ является конечномерным векторным пространством ([10], теорема 2.37).

Пусть $P = \{P_1, \ldots, P_n\} \subset \mathcal{X}, D = \sum_{M\in\mathcal{X}} a_M M, \deg(D) = \alpha$ и $\mathrm{supp}(D) \cap P = \emptyset$. Образ отображения

$$Ev_P : L(D) \longrightarrow \mathbb{F}_q^n, Ev_P(H) = (H(P_1), H(P_2), \ldots, H(P_n)) \tag{3}$$

называется АГ-кодом L-конструкции. Обозначим его через $C(\mathcal{X}, P, D_\alpha)$. Дивизор D_α будем называть дивизором кода C.

Теорема 1 ([11], теорема 4.1.1). *Пусть $\mathcal{X}$ – плоская гладкая проективная кривая рода g, $0 < \alpha < n$. Тогда АГ-код $C(\mathcal{X}, P, D_\alpha)$ является $[n, k, d]_q$-кодом, где $k \geq \alpha - g + 1$, $d \geq n - \alpha$. Если $\alpha > 2g - 2$, то $k = \alpha - g + 1$.*

Замечание 1. *В случае, когда род кривой $\mathcal{X}$ над полем $\mathbb{F}_q$ равен нулю, а $\deg(F) = 1$, т.е. $\mathcal{X}$ – проективная прямая, то АГ-код $C = C(\mathcal{X}, P, D_\alpha)$ является $[q, \alpha + 1, q - \alpha]_q$-кодом Рида-Соломона ([11], пример 4.1.5).*

2.2. Монотонность свойств c-TA и c-FP

Введём на множестве дивизоров на кривой $\mathcal{X}$ отношение $\geq$ следующим образом. Пусть D^1 и D^2 – дивизоры на кривой $\mathcal{X}$, $D^1 = \sum_{M\in\mathcal{X}} a_M M$, $D^2 = \sum_{M\in\mathcal{X}} b_M M$. Тогда положим, что $D^2 \geq D^1$, если

$$D^2 - D^1 \geq 0.$$

Теорема 2. *Пусть D^1 и D^2 – дивизоры на кривой $\mathcal{X}(F, \mathbb{F}_q)$ и $D^2 \geq D^1$. Пусть $C_i = C(\mathcal{X}, P, D^i_\alpha)$, $i = 1, 2$. Тогда 1) код $C_1 = C(\mathcal{X}, P, D^1_\alpha)$ является подкодом $C_2 = C(\mathcal{X}, P, D^2_\alpha)$, 2) $R_{TA}(C_1) \geq R_{TA}(C_2)$, 3) $R_{FP}(C_1) \geq R_{FP}(C_2)$.*

Доказательство. 1) Рассмотрим произвольный элемент $f \in \mathbb{F}_q(\mathcal{X})$. Учитывая, что $D^2 \geq D^1$, легко проверить, что если $(f) + D^1 \geq 0$, то и $(f) + D^2 \geq 0$. Тогда из определения пространства Римана-Роха вытекает, если $f \in L(D^1)$, то $f \in L(D^2)$, значит, $L(D^1) \subset L(D^2)$. Таким образом, выполняется нужное вложение.

Доказательства утверждений 2) и 3) вытекает из утверждения 1) и леммы 1. □

3. Границы для свойства c-FP

Сформулируем теорему о границах множества компрометации для c-FP-свойства.

Теорема 3. *Пусть $\mathcal{X} = \mathcal{X}(F, \mathbb{F}_q)$ – плоская гладкая проективная кривая. Рассмотрим АГ-код $C = C(\mathcal{X}, P, D_\alpha)$. Тогда*

$$R_{FP}(C) \geq \left\lceil \frac{n}{\alpha} \right\rceil.$$

Если Q – единственная бесконечная точка на $\mathcal{X}$, $|P| > 1$, а $D = \alpha Q$, то:

$$R_{FP}(C) \leq B_{FP}(C) = \left\lceil \frac{n}{\lfloor \frac{\alpha}{\deg(F)} \rfloor} \right\rceil.$$

Если род кривой $\mathcal{X}$ равен нулю, а $\deg(F) = 1$, т.е. код является кодом Рида-Соломона (см. замечание 1), то оценки в теореме превращаются в равенство $R_{FP} = \lceil n/\alpha \rceil$ из [5].

Доказательство этой теоремы проведём после нескольких вспомогательных лемм.

Лемма 2. *Рассмотрим АГ-код $C = C(\mathcal{X}(F, \mathbb{F}_q), P, D_\alpha)$. Тогда:*

$$\forall c \in \mathbb{N} \setminus \{1\}\ \forall v \in C\ \forall C_0 \in \mathrm{coal}_c(C \setminus \{v\})\ \forall \omega \in \mathrm{desc}(C_0) \setminus C_0 \ :\ |I(\omega, v)| \leq \min\{\alpha c, n\}.$$

Доказательство. Пусть $c \in \mathbb{N} \setminus \{1\}$, $v \in C$ – произвольное кодовое слово, $C_0 = \{u_1; \dots; u_c\} \in \mathrm{coal}_c(C \setminus v)$ – произвольная коалиция, $\omega \in \mathrm{desc}(C_0) \setminus C_0$ – произвольный потомок коалиции C_0. Очевидно, что $|I(\omega, v)| \leq n$. Для доказательства леммы достаточно показать, что если $\min\{\alpha c, n\} = \alpha c$, то выполняется оценка $|I(\omega, v)| \leq \alpha c$.

Теперь предположим, что $\min\{\alpha c, n\} = \alpha c$, но $|I(\omega, v)| > \alpha c$. Так как ω – потомок коалиции C_0, то для каждого номера $j \in I(\omega, v)$ найдётся такой номер $i \in \{1, \dots, c\}$, что $v_j = \omega_j = u_{ij}$. Так как мощность коалиции C_0 равна c, то существует номер $\hat{i} \in \{1, \dots, c\}$, такой, что $|I(u_{\hat{i}}, v)| > \alpha$. Тогда ввиду того, что $u_{\hat{i}}, v \in C$ получим, что:

$$d(u_{\hat{i}}, v) = n - |I(u_{\hat{i}}, v)| < n - \alpha = d^*,$$

чего не может быть согласно теореме 1. Значит, $|I(\omega, v)| \leq \alpha c$. Таким образом, $|I(\omega, v)| \leq \min\{\alpha c, n\}$. □

Для дальнейшего нам понадобятся некоторые вспомогательные конструкции. Пусть $\mathcal{X} = \mathcal{X}(F, \mathbb{F}_q)$ – плоская гладкая проективная кривая, $P = \{P_1, \dots, P_n\}$ – множество всех точек вида $P_i = (P_{i,1} : P_{i,2} : 1)$ на кривой. Назовём это множество множеством конечных точек кривой. Введём на нём два отношения эквивалентности:

$$P_i \sim_1 P_j \iff P_{i,1} = P_{j,1},\ P_i \sim_2 P_j \iff P_{i,2} = P_{j,2}.$$

Отношение $\sim_1$ разбивает P на классы эквивалентности:

$$P/\sim_1 = \{R^1, \dots, R^{k_1}\},$$

$$R^i = \{R^i_1 = (R^i_{1,1} : R^i_{1,2} : 1), R^i_2 = (R^i_{2,1} : R^i_{2,2} : 1), \dots, R^i_{l_i} = (R^i_{l_i,1} : R^i_{l_i,2} : 1)\}, \tag{4}$$

отношение $\sim_2$ разбивает P на классы эквивалентности:

$$P/\sim_2 = \{S^1, \dots, S^{k_2}\},$$

$$S^i = \{S^i_1 = (S^i_{1,1} : S^i_{1,2} : 1), S^i_2 = (S^i_{2,1} : S^i_{2,2} : 1), \dots S^i_{m_i} = (S^i_{m_i,1} : S^i_{m_i,2} : 1)\}, \tag{5}$$

где R^i_j, S^i_j – точки из P, l_i, m_i – мощности факторклассов R^i и S^i соответственно.

Значение k_1 назовём индексом множества P по первой координате, а значение k_2 индексом множества P по второй координате. Очевидно, что $k_1 \le n, k_2 \le n$. Легко проверить, если оба индекса равны 1, то множество P состоит из одной точки. Таким образом, если $|P| > 1$, то один из индексов k_1, k_2 также больше 1.

Сформулируем и докажем следующую ключевую техническую лемму, доказательство которой является громоздким.

Лемма 3. *Пусть $\mathcal{X} = \mathcal{X}(F, \mathbb{F}_q)$ – плоская гладкая проективная кривая. Рассмотрим АГ-код $C = C(\mathcal{X}(F, \mathbb{F}_q), P, D_\alpha)$, где Q – единственная бесконечная точка на $\mathcal{X}$, $|P| > 1$, $D = \alpha Q$. Тогда:*

$$\forall c \in \mathbb{N} \setminus \{1\}\ \forall v \in C\ \exists C_0 \in \mathrm{coal}_c(C \setminus \{v\})\ \exists \omega \in \mathrm{desc}(C_0) \setminus C_0\ :\ |I(\omega, v)| \ge \min\{c \left\lfloor \tfrac{\alpha}{\deg(F)} \right\rfloor, n\} \quad (6)$$

Доказательство. Предположим, что лемма доказана для $v = 0$, т.е. мы можем построить коалицию $\hat{C}_0 = \{\hat{u}_1, \dots, \hat{u}_c\}$, такую, что:

$$\forall c \in \mathbb{N} \setminus \{1\}\ \exists\ \omega \in \mathrm{desc}(\hat{C}_0) \setminus \hat{C}_0\ :\ |I(\omega, 0)| \ge \min\{c \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor, n\}.$$

Рассмотрим произвольный вектор $v \in C$, коалицию $C_0 = \{\hat{u}_1 + v, \dots, \hat{u}_c + v\}$ и вектор $\omega = \hat{\omega} + v$. Так как C – линейный код, то $C_0 \in \mathrm{coal}_c(C \setminus \{v\})$, $\omega \in \mathrm{desc}(C_0) \setminus C_0$, и выполняется:

$$|I(\omega, v)| \ge \min\{c \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor, n\}.$$

Таким образом, если лемма справедлива для $v = 0$, то она справедлива и для любых других $v \in C$.

Докажем теперь лемму в предположении, что $v = 0$. Пусть $\delta = \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor$.

Рассмотрим сначала случай, когда $c\delta < n$, и разобьём доказательство на несколько шагов.

I. Рассмотрим множество $P \subset \mathcal{X}$. Так как по построению АГ-кода $\mathrm{supp}(D) \cap P = \emptyset$, $\mathrm{supp}(D) = \{Q\}$, и Q – единственная бесконечная точка на кривой $\mathcal{X}$, то в P нет бесконечных точек, т.е. точек вида $(X_1 : X_2 : 0)$. Значит, множество P является подмножеством множества конечных точек кривой, причём, т.к. $|P| > 1$, то один из индексов P больше единицы (см. (4), (5)). Не нарушая общности, будем считать, что k_1 больше 1, и рассмотрим классы эквивалентности R^i. Перенумеруем множество P так, чтобы для первых k_1 точек из P выполняется условие: $P_i \in R^i,\ i \in \{1, \dots, k_1\}$.

II. Так как коалиция является набором кодовых векторов, каждый из которых является образом некоторой рациональной функции из пространства Римана-Роха $L(D)$ при кодирующем отображении, то для построения искомой коалиции необходимо сначала предъявить соответствующий набор рациональных функций.

Для искомых рациональных функций построим сначала вспомогательные многочлены. Рассмотрим несколько случаев.

a) Пусть $c\delta \le k_1$. Тогда в кольце $\mathbb{F}_q[x_1, x_2]$ рассмотрим следующие многочлены (см. (4)):

$$r_i(x_1, x_2) = (x_1 - R^{(i-1)\delta+1}_{1,1}) \dots (x_1 - R^{i\delta}_{1,1}),\ i \in \{1, \dots, c\} \quad (7)$$

Для каждого r_i и точки $P_l = R^{i\delta+1}_1 = (P_{l,1} : P_{l,2} : 1)$ выполняется: $r_i(P_{l,1}, P_{l,2}) \ne 0$. Степень каждого r_i равна δ.

b) Пусть теперь $k_1 < c\delta$, причём $k_1 \le \delta$ и $k_1 < c$. Тогда в кольце $\mathbb{F}_q[x_1, x_2]$ рассмотрим следующие многочлены:

$$r_i(x_1, x_2) = (x_1 - R^i_{1,1}),\ i \in \{1, \dots, k_1\}. \quad (8)$$

Для каждого r_i и точки $P_l = R_1^{i+1} = (P_{l,1} : P_{l,2} : 1)$ выполняется $r_i(P_{l,1}, P_{l,2}) \neq 0$. Степень каждого r_i не превышает δ. Рассмотрим множество ненулевых многочленов степени не выше δ, не совпадающих с r_i, $i \in \{1, \dots, k_1\}$. Таких многочленов $q^{\delta+1} - k_1 - 1$ штук. Тогда в качестве r_j, $j \in \{k_1 + 1, \dots, c\}$ возьмём многочлены из этого множества такие, что $r_j(P_{l,1}, P_{l,2}) \neq 0$ для некоторой $P_l = (P_{l,1} : P_{l,2} : 1) \in P$.

c) Пусть теперь $k_1 < c\delta$, причём $k_1 \leq \delta$, но $k_1 \geq c$. Тогда в кольце $\mathbb{F}_q[x_1, x_2]$ можно рассмотреть следующие многочлены:

$$r_i(x_1, x_2) = (x_1 - R_{1,1}^i),\ i \in \{1, \dots, c-1\},\quad r_c(x_1, x_2) = (x_1 - R_{1,1}^c) \dots (x_1 - R_{1,1}^{k_1}). \tag{9}$$

Для каждого такого r_i найдётся точка $P_l = (P_{l,1} : P_{l,2} : 1)$, для которой $r_i(P_{l,1}, P_{l,2}) \neq 0$. Для $i < c$ такой точкой, например, является точка R_1^{i+1}, а для $i = c$ такой точкой является R_1^{c-1}. Так как $k_1 - c + 1 \leq \delta - c + 1 \leq \delta$, то степень каждого r_i не превышает δ.

d) Пусть $k_1 < c\delta$, и $k_1 > \delta$. Тогда в кольце $\mathbb{F}_q[x_1, x_2]$ можно рассмотреть следующие многочлены:

$$r_i = (x_1 - R_{1,1}^{(i-1)\delta+1}) \dots (x_1 - R_{1,1}^{i\delta}),\ i \in \{1, \dots, \lceil \frac{k_1}{\delta} \rceil - 1\},$$

$$r_{\lceil \frac{k_1}{\delta} \rceil} = (x_1 - R_{1,1}^{(\lceil \frac{k_1}{\delta} \rceil - 1)\delta+1}) \dots (x_1 - R_{1,1}^{k_1}). \tag{10}$$

Для каждого такого r_i найдётся точка $P_l = (P_{l1} : P_{l2} : 1)$, такая, что $r_i(P_{l1}, P_{l2}) \neq 0$. Для $i \leq \lceil \frac{k_1}{\delta} \rceil - 1$ такой точкой, например, является точка $R_1^{i\delta+1}$, а для $i = \lceil \frac{k_1}{\delta} \rceil$ такой точкой является $R_1^{(\lceil \frac{k_1}{\delta} \rceil - 1)\delta}$. Так как $k_1 - ((\lceil \frac{k_1}{\delta} \rceil - 1)\delta + 1) + 1 < c\delta - (\lceil \frac{c\delta}{\delta} \rceil - 1)\delta \leq c\delta - (c-1)\delta \leq \delta$, то степень каждого r_i не превышает δ. В качестве r_j, $j \in \{\lceil \frac{k_1}{\delta} \rceil + 1, \dots, c\}$ возьмём произвольные ненулевые многочлены степени не выше δ, не совпадающие с r_i, $i \in \{1, \dots, \lceil \frac{k_1}{\delta} \rceil\}$, такие, что для них существует точка $(P_{l,1} : P_{l,2} : 1) \in P$ такая, что $r_i(P_{l,1}, P_{l,2}) \neq 0$.

Во всех случаях a), b), c), d) степень каждого из многочленов r_i не превышает δ, все многочлены различны и для каждого многочлена r_i найдётся такая точка $P_l = (P_{l,1} : P_{l,2} : 1) \in P$, что $r_i(P_{l,1}, P_{l,2}) \neq 0$.

Рассмотрим проективизацию построенных многочленов $r_i(x_1, x_2)$ и получим однородные многочлены $R_i(X_1, X_2, X_3)$ степени не выше δ. Построим рациональные функции:

$$H_i = \frac{R_i(X_1, X_2, X_3)}{X_3^{\deg(R_i)}}, \tag{11}$$

являющиеся элементами поля $\mathbb{F}_q(\mathcal{X})$. Покажем, что H_i принадлежит пространству Римана-Роха $L(D)$, ассоциированному с дивизором D. Действительно, согласно замечанию 2 из [7], дивизор построенной функции H_i имеет вид:

$$(H_i) = \mathcal{X} \cdot R_i(X_1, X_2, X_3) - \mathcal{X} \cdot X_3^{\deg(R_i)}.$$

В силу замечания к теореме 2.23 из [10]

$$\sum_{M \in \mathcal{X}(F, \mathbb{F}_q)} I(M; \mathcal{X}(F, \mathbb{F}_q); G) \leq \deg(G) \cdot \deg(F),$$

поэтому в силу (2)

$$\mathcal{X} \cdot X_3^{\deg(R_i)} = \textstyle\sum_{M \in \mathcal{X}(F, \mathbb{F}_q)} I(M; \mathcal{X}(F, \mathbb{F}_q); X_3^{\deg(R_i)})M = I(Q; \mathcal{X}(F, \mathbb{F}_q); X_3^{\deg(R_i)})Q$$
$$\leq \deg(X_3^{\deg(R_i)})\deg(F)Q = \deg(R_i)\deg(F)Q.$$

Тогда

$$(H_i) = \mathcal{X} \cdot R_i(X_1, X_2, X_3) - \mathcal{X} \cdot X_3^{\deg(R_i)} \geq \sum_{P_j \in P} I(P_j, X, R_i)P_j - \deg(R_i)\deg(F)Q,$$

$$(H_i) + D \geq \sum_{P_j \in P} I(P_j, \mathcal{X}, R_i)P_j - \deg(R_i)\deg(F)Q + \alpha Q =$$

$$= \sum_{P_j \in P}^{\delta i} I(P_j, \mathcal{X}, \mathcal{P})P_j + (\alpha - \deg(R_i)\deg(F))Q.$$

Так как $\deg(R_i) \leq \delta = \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor$, то

$$(H_i) + D \geq \sum_{P_j \in P} I(P_j, \mathcal{X}, \mathcal{P})P_j + (\alpha - \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor \deg(F))Q \geq 0,$$

значит, $H_i \in L(D)$.

III. Построим теперь искомую коалицию $C_0 = \{u_1; \dots; u_c\}$ следующим образом:

$$u_i = Ev_P(H_i) = (H_i(P_1), \dots, H_i(P_n)). \tag{12}$$

Все многочлены r_i различны, поэтому и все R_i, а также H_i тоже различны. Для каждого r_i существует точка $P_l = (P_{l,1} : P_{l,2} : 1) \in P$ такая, что $r_i(P_{l,1}, P_{l,2}) \neq 0$, следовательно, и $H_i(P_{l,1} : P_{l,2} : 1) \neq 0$. Таким образом, в коалиции ровно c различных ненулевых векторов.

IV. Теперь для каждого из рассмотренных на шаге II случаев a)–d) построим искомого потомка ω.

В случае a) коалиция C_0 (см. (12)) в силу (7) и (11) выглядит следующим образом:

$$\begin{cases} u_1 = (0, \dots, 0, H(P_{\delta+1}), \dots, H(P_n)) \\ \dots \\ u_i = (H(P_1), \dots, H(P_{\delta(i-1)}), 0, \dots, 0, H(P_{\delta i+1}), \dots, H(P_n)) \\ \dots \\ u_c = (H(P_1), \dots, H(P_{\delta(c-1)}), 0, \dots, 0, H(P_{\delta c+1}), \dots, H(P_n)) \end{cases},$$

Рассмотрим потомка коалиции C_0:

$$\omega = (0, \dots, 0, \omega_{\delta c+1}, \dots, \omega_n),$$

где для каждого $j \in \{\delta c + 1, \dots, n\}$ значение ω_j задаётся как произвольный элемент из $\{u_{1,j}, \dots, u_{c,j}\}$. Ясно, что $\omega \in \mathrm{desc}(C_0) \setminus C_0$. По построению:

$$|I(\omega, 0)| \geq \delta c = c \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor.$$

В случае b) коалиция в силу (8) и (11) выглядит следующим образом:

$$\begin{cases} u_1 = (0, \dots, H(P_i), \dots, H(P_n)) \\ \dots \\ u_i = (H(P_1), \dots, H(P_{i-1}), 0, H(P_{i+1}), \dots, H(P_n)) \\ \dots \\ u_{k_1} = (H(P_1), \dots, H(P_{k_1-1}), 0, H(P_{k_1+1}), \dots, H(P_n)) \\ \dots \\ u_j = (H(P_1), \dots, H(P_{j-1}), H(P_j), H(P_{j+1}), \dots, H(P_n)) \\ \dots \\ u_c = (H(P_1), \dots, H(P_{c-1}), H(P_{c+1}), \dots, H(P_n)) \end{cases}$$

Построим потомка ω следующим образом. В качестве ω_i, где $1 \le i \le k_1$, из вектора u_i возьмём ноль, стоящий там на i-ой позиции. Заметим, что для любой позиции j такой, что $j > k_1$, точка P_j лежит в каком-либо классе эквивалентности R^m. Тогда $u_{m,j} = 0$, т.к. по построению значение H_m равно нулю на любой точке из R^m, в том числе на P_j. Значит, для любой такой позиции j мы можем выбрать $\omega_j = u_{m,j} = 0$. Таким образом, комбинированием только первых k_1 векторов мы можем выбрать потомка ω, совпадающего с нулевым вектором. Тогда:

$$|I(\omega, 0)| = n.$$

В случае c) коалиция в силу (9) и (11) выглядит следующим образом:

$$\begin{cases} u_1 = (0, \dots, H(P_i), \dots, H(P_n)) \\ \dots \\ u_i = (H(P_1), \dots, H(P_{i-1}), 0, H(P_{i+1}), \dots, H(P_n)) \\ \dots \\ u_c = (H(P_1), \dots, H(P_{c-1}), 0, \dots, 0, H(P_{k_1}), \dots, H(P_n)) \end{cases}$$

Построим потомка ω. В качестве ω_i, где $1 \le i \le c$, из вектора u_i возьмём ноль, стоящий там на i-ой позиции. Если $c \le i \le k_1$, то в качестве элемента на позиции i возьмём ноль из вектора u_c, также стоящий там на i-ой позиции. Аналогично предыдущему случаю, для любой позиции j такой, что $j > k_1$, точка P_j лежит в одном из классов эквивалентности R^m. Тогда $u_{m,j} = 0$, т.к. значение H_m равно нулю на любой точке из R^m. Значит, для любой такой позиции j мы можем выбрать $\omega_j = u_{m,j} = 0$. Комбинированием всех c векторов мы можем выбрать потомка ω, совпадающего с нулевым вектором:

$$|I(\omega, 0)| = n.$$

В случае d) коалиция в силу (10) и (11) выглядит следующим образом:

$$\begin{cases} u_1 = (0, \dots, 0, H(P_{\delta+1}), \dots, H(P_n)) \\ \dots \\ u_i = (H(P_1), \dots, H(P_{\delta(i-1)}), 0, \dots, 0, H(P_{\delta i+1}), \dots, H(P_n)) \\ \dots \\ u_{\lceil \frac{k_1}{\delta} \rceil} = (H(P_1), \dots, H(P_{(\lceil \frac{k_1}{\delta} \rceil - 1)\delta,1}), 0, \dots, 0, H(P_{k_1}), \dots, H(P_n)) \\ \dots \\ u_c = (H(P_1), \dots, H(P_{c-1}), H(P_{c+1}), \dots, H(P_n)) \end{cases}$$

Построим потомка ω в этом случае. В качестве ω_i, где $1 \le i \le \lceil \frac{k_1}{\delta} \rceil$, из вектора u_i возьмём ноль, стоящий там на i-ой позиции. Если $\lceil \frac{k_1}{\delta} \rceil \le i \le k_1$, то в качестве элемента на позиции i возьмём ноль из вектора $u_{\lceil \frac{k_1}{\delta} \rceil}$, также стоящий там на i-ой позиции. Аналогично предыдущему случаю, для любой позиции j такой, что $j > k_1$, существует номер m такой, что $u_{m,j} = 0$. Значит, для любой такой позиции j мы можем выбрать $\omega_j = u_{m,j} = 0$. Тогда комбинированием первых $\lceil \frac{k_1}{\delta} \rceil$ векторов мы можем выбрать потомка ω, совпадающего с нулевым вектором:

$$|I(\omega, 0)| = n.$$

Итак, во всех вариантах, когда $c\delta < n$, найдётся такой потомок ω коалиции C_0, что $|I(\omega, 0)| \ge c\delta = \min\{c\delta, n\}$.

Таким образом, лемма в случае, когда $c\delta < n$, доказана. Теперь рассмотрим случай, когда $c\delta \geq n$.

Построим коалицию C_0 в этом случае. Пусть $\hat{c} = \lfloor \frac{n}{\delta} \rfloor$, тогда $\hat{c}\delta < n$. Набор многочленов $r_i, i \in \{1, \dots, \hat{c}\}$ и первые $\hat{c}$ элементов коалиции построим так же, как это было описано выше для случая $c\delta < n$ на шагах II и III. Теперь нужно достроить коалицию до необходимой мощности c.

Если на шаге II для $\hat{c}$ реализовались случаи b), c) или d), то, как показано на шаге IV, в качестве потомка построенной коалиции мощности $\hat{c}$ уже может быть выбран нулевой вектор. Поэтому в качестве остальных $c - \hat{c}$ членов коалиции можно взять любые из оставшихся ненулевых кодовых векторов.

Предположим, что на шаге II для $\hat{c}$ реализовался случай a). Если $\hat{c} = \lfloor \frac{n}{\delta} \rfloor = \lceil \frac{n}{\delta} \rceil$, то дополнительных построений проводить не нужно, т.к. в качестве потомка построенной коалиции мощности $\hat{c}$ уже может быть выбран нулевой вектор, и в качестве остальных членов коалиции можно взять любые из оставшихся ненулевых кодовых векторов. Если $\hat{c} = \lfloor \frac{n}{\delta} \rfloor < \lceil \frac{n}{\delta} \rceil$, то построим элемент коалиции с номером $\lceil \frac{n}{\delta} \rceil$ следующим образом. Возьмём проективизацию $R_{\lceil \frac{n}{\delta} \rceil}$ многочлена $r_{\lceil \frac{n}{\delta} \rceil} = (x_1 - P_{(\lceil \frac{n}{\delta} \rceil - 1)\delta + 1,1}) \dots (x_1 - P_{n,1})$ и поделим на X_3^δ, получив рациональную функцию $H_{\lceil \frac{n}{\delta} \rceil}$. Очевидно, что значение $H_{\lceil \frac{n}{\delta} \rceil}$ на точках $P_{(\lceil \frac{n}{\delta} \rceil - 1)\delta + 1}, \dots, P_n$ равно нулю. Аналогично показанному на шаге II в случае $c\delta < n$, проверяется, что $H_{\lceil \frac{n}{\delta} \rceil} \in L(D)$. Тогда можно построить очередной член коалиции $u_{\lceil \frac{n}{\delta} \rceil}$, являющийся образом функции $H_{\lceil \frac{n}{\delta} \rceil}$. По построению на позициях $(\lceil \frac{n}{\delta} \rceil - 1)\delta + 1, \dots, n$ в $u_{\lceil \frac{n}{\delta} \rceil}$ находятся нули. В этом случае оставшиеся $c - \lceil \frac{n}{\delta} \rceil$ членов коалиции выберем как произвольные кодовые ненулевые слова, не совпадающие с построенными ранее членами коалиции. В качестве потомка построенной коалиции может быть выбран нулевой вектор. Действительно, первые $\hat{c}\delta$ позиций могут быть заполнены нулями аналогично случаю, когда $\hat{c} = \lfloor \frac{n}{\delta} \rfloor = \lceil \frac{n}{\delta} \rceil$, а остальные $n - \hat{c}\delta$ позиций можно заполнить нулями, стоящими на соответствующих позициях в векторе $u_{\lceil \frac{n}{\delta} \rceil}$.

Таким образом, построена коалиция $C_0 \in \mathrm{coal}_c(C \setminus \{0\})$. В качестве $\omega \in \mathrm{desc}(C_0) \setminus C_0$ можно выбрать нулевой вектор. Тогда $|I(\omega, 0)| = n \geq n = \min\{c\delta, n\}$.

Итак, лемма доказана и в случае, когда $c\delta \geq n$. □

Доказательство предыдущей леммы довольно громоздко, но содержит описание способа построения по заданной мощности c и кодовому вектору v коалиции C_0 и такого её потомка ω, который совпадает с v не менее, чем в $\min\{c\delta, n\}$ позициях. Этот способ важен при анализе стойкости схем широковещательного шифрования. Проиллюстрируем его на примерах.

Пример 1. *Пусть $\alpha = 7$, $c = 2$. Рассмотрим кривую $\mathcal{X}$ рода $g = 1$, заданную следующим многочленом:*

$$F(X_1, X_2, X_3) = X_2^2 X_3 + X_1 X_2 X_3 + X_2 X_3^2 - X_1^3 - X_3^3$$

над полем $\mathbb{F}_8 = \mathbb{F}_2[\xi]/(\xi^3 + \xi + 1)$. Тогда $\delta = \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor = \lfloor \frac{7}{3} \rfloor = 2$. Выпишем все точки кривой:

$$Q = (0 : 1 : 0), P_1 = (1 : 0 : 1), P_2 = (\xi : \xi : 1), P_3 = (\xi^2 : \xi^2 : 1),$$

$$P_4 = (\xi^3 : \xi^4 : 1), P_5 = (\xi^4 : \xi^4 : 1), P_6 = (\xi^5 : \xi : 1), P_7 = (\xi^6 : \xi : 1), P_8 = (\xi^4 : 1 : 1),$$

$$P_9 = (\xi^5 : \xi^2 : 1), P_{10} = (\xi^6 : \xi^4 : 1), P_{11} = (\xi^2 : 1 : 1), P_{12} = (\xi^3 : \xi^2 : 1), P_{13} = (\xi : 1 : 1).$$

Рассмотрим АГ-код L-конструкции $C = C(\mathcal{X}, \{P_1, \dots, P_{13}\}, D = \alpha Q)$ и кодовый вектор $v = 0$.

Построим искомые коалицию C_0 этого кода мощности c и потомка ω, используя лемму 3. Классы эквивалентности по первой координате выглядят следующим образом:

$$R^1 = \{P_1\}, R^2 = \{P_2, P_{13}\}, R^3 = \{P_3, P_{11}\}, R^4 = \{P_4, P_{12}\},$$

$$R^5 = \{P_5, P_8\}, R^6 = \{P_6, P_9\}, R^7 = \{P_7, P_{10}\}.$$

Значит, $k_1 = 7 > 1$ и при построении искомой коалиции можно использовать классы R^i. Нумерация точек соответствует наложенному в лемме требованию, что $P_i \in R^i, i \in \{1, \dots, k_1\}$. В нашем случае $c\delta = 2 \cdot 2 < n = 13$ и $c\delta = 2 \cdot 2 \leq k_1 = 7 \leq n = 13$. Такому набору параметров соответствует случай а) на шаге II из леммы 3. Значит, мы можем построить коалицию C_0 и потомка ω такого, что $I(\omega, 0) \geq c\delta = 4$. Многочлены r_i выглядят следующим образом:

$$r_1 = (x - R^1_{1,1})(x - R^2_{1,1}) = (x - P_{1,1})(x - P_{2,1}) = (x - 1)(x - \xi) = x^2 - \xi^3 x + \xi,$$

$$r_2 = (x - R^3_{1,1})(x - R^4_{1,1}) = (x - P_{3,1})(x - P_{4,1}) = (x - \xi^2)(x - \xi^3) = x^2 - \xi^5 x + \xi^5,$$

а R_i выглядят так:

$$R_1 = X_1^2 - \xi^3 X_1 X_3 + \xi X_3^2,$$
$$R_2 = X_1^2 - \xi^5 X_1 X_3 + \xi^5 X_3^2.$$

Тогда H_i выглядят следующим образом:

$$H_1 = \frac{X_1^2 - \xi^3 X_1 X_3 + \xi X_3^2}{X_3^2},\ H_2 = \frac{X_1^2 - \xi^5 X_1 X_3 + \xi^5 X_3^2}{X_3^2}.$$

Из замечания 2 в [7] и теоремы 2.23 в [10] вычисляем:

$$(H_1) = 2P_1 + P_2 + P_{13} - 4Q, (H_2) = P_3 + P_4 + P_{11} + P_{12} - 4Q.$$

Тогда

$$(H_1) + D = (H_1) + 7Q = 2P_1 + P_2 + P_{13} - 4Q + 7Q = 2P_1 + P_2 + P_{13} + 3Q \geq 0,$$

$$(H_2) + D = (H_2) + 7Q = P_3 + P_4 + P_{11} + P_{12} - 4Q + 7Q = P_3 + P_4 + P_{11} + P_{12} + 3Q \geq 0,$$

значит, H_1 и H_2 принадлежат пространству Римана-Роха $L(D)$. Отображение $Ev_P(L(D))$ переведёт H_1 в вектор $(0, 0, H_1(P_3), H_1(P_4), \dots, H_1(P_{13}))$, а H_2 в вектор $(H_2(P_1), H_2(P_2), 0, 0, H_2(P_5), \dots, H_2(P_{13}))$. Коалиция из этих двух векторов гарантированно генерирует потомка ω с $c\delta = 4$ нулями на первых четырёх позициях. Тогда $I(\omega, 0) \geq c\delta = 4$. Искомая коалиция C_0 построена.

Пример 2. *Пусть $\alpha = 5, c = 2$. Рассмотрим кривую $\mathcal{X}$ рода $g = 0$, заданную следующим многочленом:*

$$F(X_1, X_2, X_3) = X_2 - X_3 = 0$$

над полем $\mathbb{F}_8 = \mathbb{F}_2[\xi]/(\xi^3 + \xi + 1)$. Обозначим $\delta = \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor = \lfloor \frac{5}{1} \rfloor = 5$. Выпишем все точки кривой:

$$Q = (1 : 0 : 0), P_1 = (0 : 1 : 1), P_2 = (1 : 1 : 1), P_3 = (\xi : 1 : 1),$$

$$P_4 = (\xi^2 : 1 : 1), P_5 = (\xi^3 : 1 : 1), P_6 = (\xi^4 : 1 : 1), P_7 = (\xi^5 : 1 : 1), P_8 = (\xi^6 : 1 : 1).$$

Рассмотрим АГ-код L-конструкции $C = C(\mathcal{X}, \{P_1, \dots, P_8\}, D = \alpha Q)$ и кодовый вектор $v = 0$. Отметим, что согласно замечанию 1, этот код является кодом Рида-Соломона.

Построим коалицию C_0 этого кода мощности c, используя алгоритм из леммы 3. Классы эквивалентности по первой координате построим следующим образом: $R^i = \{P_i\}, i = 1, \dots, 8$. Индекс k_1 равен 8. Отметим, что индекс по второй координате равен 1. Легко видеть, что по построению классов необходимая далее перенумерация уже произведена. В нашем случае $c\delta = 2 \cdot 5 = 10 \geq n = 8$. Такому набору параметров соответствует второй случай из леммы 3. В этом случае мы можем построить коалицию C_0 и потомка ω такого, что $I(\omega, 0) = n = 8$. По алгоритму сначала мы должны построить

$\hat{c} = \lceil n/\delta \rceil = 1$ *многочленов по алгоритму из шага II, заменяя* c *на* $\hat{c}$*. В этом случае для* $\hat{c}$ *реализуется случай а) из шага II, когда* $\hat{c}\delta \le k_1 \le n$*. Многочлен* r_1 *выглядит следующим образом:*

$$r_1 = (x - R^1_{1,1})(x - R^2_{1,1})(x - R^3_{1,1})(x - R^4_{1,1})(x - R^5_{1,1}) =$$

$$= (x - P_{1,1})(x - P_{2,1})(x - P_{3,1})(x - P_{4,1})(x - P_{5,1}) =$$

$$= (x - 0)(x - 1)(x - \xi)(x - \xi^2)(x - \xi^3) = x^5 + \xi^2 x^4 + \xi^5 x^3 + \xi^5 x^2 + \xi^6 x,$$

а R_1 *выглядит так:*

$$R_1 = X_1^5 + \xi^2 X_1^4 X_3 + \xi^5 X_1^3 X_3^2 + \xi^5 X_1^2 X_3^3 + \xi^6 X_1 X_3^4.$$

Тогда H_1 *выглядит следующим образом:*

$$H_1 = \frac{X_1^5 + \xi^2 X_1^4 X_3 + \xi^5 X_1^3 X_3^2 + \xi^5 X_1^2 X_3^3 + \xi^6 X_1 X_3^4}{X_3^5}.$$

Из замечания 2 в [7] и теоремы 2.23 в [10] вычисляем:

$$(H_1) = P_1 + P_2 + P_3 + P_4 + P_5 - 5Q.$$

Тогда

$$(H_1) + D = (H_1) + 5Q = P_1 + P_2 + P_3 + P_4 + P_5 - 5Q + 5Q = P_1 + P_2 + P_3 + P_4 + P_5 \ge 0$$

и $H_1 \in L(D)$*. Далее строим* r_2 *по оставшимся* $n - \hat{c} = 3$ *нулям:*

$$r_2 = (x - P_{6,1})(x - P_{7,1})(x - P_{8,1}) = (x - \xi^4)(x - \xi^5)(x - \xi^6) = x^3 + \xi^2 x^2 + x + \xi.$$

Многочлен R_2 *выглядит так:*

$$R_2 = X_1^3 + \xi^2 X_1^2 X_3 + X_1 X_3^2 + \xi X_3^3,$$

а H_2 *выглядит так:*

$$\frac{X_1^3 + \xi^2 X_1^2 X_3 + X_1 X_3^2 + \xi X_3^3}{X_3^3}.$$

Вычислим $(H_2) + D = P_6 + P_7 + P_8 - 3Q + 5Q \ge 0$*, тогда* $H_2 \in L(D)$*. Отображение* $Ev_P(L(D))$ *переведёт* H_1 *в вектор* $(0, 0, 0, 0, 0, H_1(P_6), H_1(P_7), H_1(P_8))$*, а* H_2 *в вектор* $(H_2(P_1), H_2(P_2), H_2(P_3), H_2(P_4), H_2(P_5), 0, 0, 0)$*. Тогда коалиция из этих двух векторов гарантированно генерируют потомка* $\omega = 0$*: возьмём первые пять нулей из первого вектора, а последние три нуля – из второго. Получаем* $I(\omega, 0) \ge c\delta = n = 8$*. Искомая коалиция* C_0 *построена.*

Доказательство теоремы 3.

1. Докажем сначала, что

$$R_{FP}(C) \ge \left\lceil \frac{n}{\alpha} \right\rceil.$$

Для того, чтобы проверить это неравенство, достаточно показать, что если $c < \left\lceil \frac{n}{\alpha} \right\rceil$, то код C обладает c-FP свойством.

Пусть $c < \left\lceil \frac{n}{\alpha} \right\rceil$, тогда $c < \frac{n}{\alpha}$, и $c\alpha < n$. В силу леммы 2 отсюда получаем:

$$\forall v \in C \; \forall C_0 \in \mathrm{coal}_c(C \setminus \{v\}) \; \forall \omega \in \mathrm{desc}(C_0) \setminus C_0 \; : \; |I(\omega, v)| \le c\alpha < n.$$

Значит,

$$\forall v \in C \; \forall C_0 \in \mathrm{coal}_c(C \setminus \{v\}) \; \forall \omega \in \mathrm{desc}(C_0) \setminus C_0 \; : \; v \ne \omega,$$

это и означает, что C является c-FP кодом.

2. Теперь докажем, что если Q – единственная бесконечная точка на кривой $\mathcal{X}$, $|P| > 1$, а $D = \alpha Q$, то

$$R_{FP}(C) \le B_{FP}(C) = \left\lceil \frac{n}{\lfloor \frac{\alpha}{\deg(F)} \rfloor} \right\rceil.$$

Пусть $\hat{c}$ — произвольное целое такое, что $\hat{c} \ge B_{FP}$. Чтобы доказать искомую оценку, достаточно показать, что при числе злоумышленников $\hat{c}$ рассматриваемое FP-свойство не выполнено. В силу (6) из леммы 3:

$$\forall c \in \mathbb{N} \setminus \{1\}\ \forall v \in C\ \exists C_0 \in \mathrm{coal}_c(C \setminus \{v\})\ \exists \omega \in \mathrm{desc}(C_0) \setminus C_0\ :$$

$$|I(\omega, v)| \ge \min\{c \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor, n\}.$$

По предположению $\hat{c} \ge B_{FP}$, поэтому

$$|I(\omega, v)| \ge \min\{c \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor, n\} = n.$$

То есть $|I(\omega, v)| = n$, значит, $\omega = v \in \mathrm{desc}(C_0) \setminus C_0$. Это и значит, что FP-свойство не выполнено, следовательно, $R_{FP}(C) \le B_{FP}(C)$.

Теорема 3 доказана.

4. Границы для свойства c-ТА

Сформулируем теорему о границах свойства c-ТА.

Теорема 4. *Пусть $\mathcal{X} = \mathcal{X}(F, \mathbb{F}_q)$ – плоская гладкая проективная кривая. Рассмотрим АГ-код $C = C(\mathcal{X}(F, \mathbb{F}_q), P, D_\alpha)$. Тогда*

$$R_{TA}(C) \ge \lceil \sqrt{n/\alpha} \rceil.$$

Если Q – единственная бесконечная точка на $\mathcal{X}$, $|P| > 1$, $D = \alpha Q$, то:

$$R_{TA}(C) \le B_{TA}(C) = \left\lceil \frac{n + \alpha}{2 \left\lfloor \frac{\alpha}{\deg(F)} \right\rfloor} \right\rceil.$$

Доказательство. Первое утверждение доказано в теореме 1 из [7].

Докажем второе утверждение. Пусть, как и выше в лемме 3, $\delta = \lfloor \frac{\alpha}{\deg(F)} \rfloor$. Для доказательства этого утверждения достаточно показать, что если c – произвольное целое число такое, что

$$c \ge B_{TA}(C) = \left\lceil \frac{n + \alpha}{2\delta} \right\rceil,$$

то C не является c-ТА кодом. Тогда в предположении, что $c \ge B_{TA}(C)$, получаем:

$$c\delta \ge (n + \alpha)/2. \tag{13}$$

Пусть $v \in C$ – произвольное кодовое слово. Согласно определению c-ТА для того, чтобы подтвердить, что это свойство не выполнено, достаточно показать:

$$\exists C_0 \in \mathrm{coal}_c(C)\ \exists \omega \in \mathrm{desc}(C_0)\ \ :\ \forall i \in \{1, \dots, c\}\ |I(\omega, u_i)| \le |I(\omega, v)|.$$

В качестве C_0 и ω рассмотрим построенные в лемме 3 коалицию C_0 и её потомка ω и покажем, что выполняются искомые неравенства. Не нарушая общности предположим, что $k_1 > 1$, и для построения коалиции C_0 использовались классы эквивалентности R^i (см. (4), (5)).

Рассмотрим два случая.

1. Пусть $c\delta \le k_1$. Тогда в силу леммы 3

$$|I(\omega, v)| \ge \min\{c\delta, n\} = c\delta, \tag{14}$$

т.к. $k_1 \le n$. Для произвольного $S \subset \{1, \dots, n\}$ определим $I_S(u, v) = \{i \in S : u_i = v_i\}$. Пусть $A = \{1, \dots, c\delta\}$. Покажем, что

$$\forall i \in \{1, \dots, c\} \; |I_A(\omega, u_i)| \le \alpha. \tag{15}$$

Предположим противное, т.е. найдётся такой номер i_0, что $|I_A(\omega, u_{i_0})| > \alpha$. Тогда существует $r > \alpha$ позиций из A таких, что в каждой такой позиции k выполняется $u_{i_0,k} = \omega_k$. Согласно построению из леммы 3 (см. шаг IV, случай а)), $\omega_j = v_j$ для всех $j \in A$, поэтому $u_{i_0,k} = v_k$. Значит, $|I(v, u_{i_0})| = r > \alpha$. Тогда

$$d(v, u_{i_0}) = n - |I(v, u_{i_0})| < n - \alpha,$$

чего в силу теоремы 1 быть не может, значит, (15) выполнено.

Докажем неравенство

$$|I(\omega, u_i)| \le n - c\delta + \alpha.$$

Ввиду того, что

$$I(\omega, u_i) = I_A(\omega, u_i) \cup I_{\{1,\dots,n\}\setminus A}(\omega, u_i),$$

и неравенства (15) получаем, что

$$|I(\omega, u_i)| = |I_A(\omega, u_i)| + |I_{\{1,\dots,n\}\setminus A}(\omega, u_i)| \le \alpha + n - c\delta. \tag{16}$$

Из (13) вытекает, что

$$n - c\delta + \alpha \le c\delta.$$

Тогда учитывая сначала (16), а потом (14), получим:

$$|I(\omega, u_i)| \le n - c\delta + \alpha \le c\delta \le |I(\omega, v)|.$$

Это означает, что свойство c-TA при данных условиях не выполнено.

2. Пусть $c\delta > k_1$. В случае, когда $k_1 < c\delta < n$, в лемме 3 показано, что $|I(\omega, v)| = n$, т.е. $\omega = v$. Это означает, что C не является c-FP кодом. Аналогично, в случае $c\delta \ge n$ из второго утверждения в теореме 3 вытекает, что код C не является c-FP кодом. Значит, при $c\delta > k_1$ код C не является и c-TA кодом (см. (1)).

Таким образом, показано, что если c – произвольное целое такое, что $c \ge B_{TA}(C)$, то нарушается рассматриваемое TA-свойство. Теорема доказана. □

Если род кривой $\mathcal{X}$ равен нулю, а $\deg(F) = 1$, т.е. код является кодом Рида-Соломона (см. замечание 1), то оценки в теореме превращаются в оценки из [5]:

$$\left\lceil \sqrt{\frac{n}{k-1}} \right\rceil \le R_{TA} \le \left\lfloor \frac{n+k-1}{2(k-1)} \right\rfloor.$$

References

[1] D. R. Stinson and R. Wei, "Combinatorial properties and constructions of traceability schemes and frameproof codes", *SIAM Journal on Discrete Mathematics*, vol. 11, no. 1, pp. 41–53, 1998.

[2] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes", *Information Theory, IEEE Transactions*, vol. 47, no. 3, pp. 1042–1049, 2001.

[3] A. Silverberg, J. Staddon, and J. Walker, "Applications of list decoding to tracing traitors", *Information Theory, IEEE Transactions*, vol. 49, no. 5, pp. 1312–1318, 2003.

[4] G. A. Kabatyansky, "Traceability codes and their generalizations", *Problems of Information Transmission*, vol. 55, no. 3, pp. 93–105, 2019.

[5] V. M. Deundyak and V. V. Mkrtichyan, "Issledovaniye granits primeneniya skhemy zashchity informatsii, osnovannoy na RS-kodakh", *Diskretn. Anal. Issled. Oper.*, vol. 18, no. 3, pp. 21–38, 2011.

[6] V. M. Deundyak, S. A. Yevpak, and V. V. Mkrtichyan, "Analysis of properties of q-ary Reed–Muller error-correcting codes viewed as codes for copyright protection", *Problems of Information Transmission*, vol. 51, no. 4, pp. 398–408, 2015.

[7] D. V. Zagumennov and V. V. Mkrtichyan, "On application of algebraic geometry codes of L-construction in copy protection", *Prikladnaya Diskretnaya Matematika*, vol. 44, pp. 67–93, 2019.

[8] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977, vol. 16.

[9] S. A. Evpak and V. V. Mkrtichyan, "Usloviya primeneniya q-ichnyh kodov Rida–Mallera v special'nyh skhemah zashchity informacii ot nesankcionirovannogo dostupa", *Vladikavk. matem. zhurn.*, vol. 16, no. 2, pp. 38–45, 2014.

[10] T. Høholdt, J. H. van Lint, and R. Pellikaan, "Algebraic geometry codes", *Handbook of coding theory*, vol. 1, no. Part 1, pp. 871–961, 1998.

[11] S. G. Vladets, D. Y. Nogin, and M. A. Tsfasman, *Algebrogeometricheskie kody. Osnovnye ponyatiya*. MCCME, 2003.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

THEORY OF COMPUTING

On a Segment Partition for Entropy Estimation

E. A. Timofeev[1]

DOI: 10.18255/1818-1015-2020-1-40-47

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 94A17
Research article
Full text in Russian

Received November 23, 2019
After revision February 18, 2020
Accepted February 28, 2020

Let Q_n be a partition of the interval $[0,1]$ defines as

$$Q_1 = \{0, q^2, q, 1\}.$$
$$Q'_{n+1} = qQ_n \cap q^2Q_n, \quad Q''_{n+1} = q^2 + qQ_n \cap qQ_n, \quad Q'''_{n+1} = q^2 + qQ_n \cap q + q^2Q_n,$$
$$Q_{n+1} = Q'_{n+1} \cup Q''_{n+1} \cup Q'''_{n+1},$$

where $q^2 + q = 1$.
The sequence $d = 1, 2, 1, 0, 1, 2, 1, 0, 1, 0, 1, 2, 1, 0, 1, 2, 1, \ldots$ defines as follows.

$$d_1 = 1,\ d_2 = 2,\ d_4 = 0;$$
$$d[2F_{2n} + 1 : 2F_{2n+1} + 1] = d[1 : 2F_{2n-1} + 1];$$
$$n = 0, 1, 2, \ldots;$$
$$d[2F_{2n+1} + 2 : 2F_{2n+1} + 2F_{2n-2}] = d[2F_{2n-1} + 2 : 2F_{2n}];$$
$$d[2F_{2n+1} + 2F_{2n-2} + 1 : 2F_{2n+1} + 2F_{2n-1} + 1] = d[1 : 2F_{2n-3} + 1];$$
$$d[2F_{2n+1} + 2F_{2n-1} + 2 : 2F_{2n+2}] = d[2F_{2n-1} + 2 : 2F_{2n}];$$
$$n = 1, 2, 3, \ldots;$$

where F_n are Fibonacci numbers ($F_{-1} = 0, F_0 = F_1 = 1$).
The main result of this paper.
Theorem.

$$Q'_n = 1 - Q'''_n = \left\{ \sum_{i=1}^{k} q^{n+d_i},\ k = 0, 1, \ldots, m_n \right\},$$

$$Q''_n = 1 - Q''_n = \left\{ q^2 + \sum_{i=m_n}^{k} q^{n+d_i},\ k = m_n - 1, m_n, \ldots, m_{n+1} \right\},$$

where $m_{2n} = 2F_{2n-2},\ m_{2n+1} = 2F_{2n-1} + 1$.

Keywords: measure; metric; entropy; estimation; unbiased; self-similarity; Bernoulli measure

INFORMATION ABOUT THE AUTHORS

Evgeniy Alexandrovich Timofeev | orcid.org/0000-0002-3094-4390. E-mail: timofeevEA@gmail.com
Sc.D., professor.

For citation: E. A. Timofeev, "On a Segment Partition for Entropy Estimation", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 40-47, 2020.

© Timofeev E. A., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

THEORY OF COMPUTING

Об одном разбиении отрезка, применяемом для оценки энтропии

Е. А. Тимофеев[1]

DOI: 10.18255/1818-1015-2020-1-40-47

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003 Россия.

УДК 519.17
Научная статья
Полный текст на русском языке

Получена 23 ноября 2019 г.
После доработки 18 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

В работе изучается разбиение отрезка, которое строится по следующему правилу:

$$Q_1 = \{0, q^2, q, 1\}.$$
$$Q'_{n+1} = qQ_n \cap q^2Q_n, \quad Q''_{n+1} = q^2 + qQ_n \cap qQ_n, \quad Q'''_{n+1} = q^2 + qQ_n \cap q + q^2Q_n,$$
$$Q_{n+1} = Q'_{n+1} \cup Q''_{n+1} \cup Q'''_{n+1},$$

где $q^2 + q = 1$.
Введем последовательность чисел $d = 1, 2, 1, 0, 1, 2, 1, 0, 1, 0, 1, 2, 1, 0, 1, 2, 1, \dots$, положив

$$d_1 = 1, \ d_2 = 2, \ d_4 = 0;$$
$$d[2F_{2n} + 1 : 2F_{2n+1} + 1] = d[1 : 2F_{2n-1} + 1];$$
$$n = 0, 1, 2, \dots;$$
$$d[2F_{2n+1} + 2 : 2F_{2n+1} + 2F_{2n-2}] = d[2F_{2n-1} + 2 : 2F_{2n}];$$
$$d[2F_{2n+1} + 2F_{2n-2} + 1 : 2F_{2n+1} + 2F_{2n-1} + 1] = d[1 : 2F_{2n-3} + 1];$$
$$d[2F_{2n+1} + 2F_{2n-1} + 2 : 2F_{2n+2}] = d[2F_{2n-1} + 2 : 2F_{2n}];$$
$$n = 1, 2, 3, \dots;$$

где F_n – числа Фибоначчи ($F_{-1} = 0, F_0 = F_1 = 1$).
Основной результат работы.
Теорема.

$$Q'_n = 1 - Q'''_n = \left\{ \sum_{i=1}^{k} q^{n+d_i}, \ k = 0, 1, \dots, m_n \right\},$$

$$Q''_n = 1 - Q''_n = \left\{ q^2 + \sum_{i=m_n}^{k} q^{n+d_i}, \ k = m_n - 1, m_n, \dots, m_{n+1} \right\},$$

где $m_{2n} = 2F_{2n-2}, \ m_{2n+1} = 2F_{2n-1} + 1$.

Ключевые слова: мера; метрика; энтропия; оценка; несмещенность; самоподобие; мера Бернулли

ИНФОРМАЦИЯ ОБ АВТОРАХ

Евгений Александрович Тимофеев | orcid.org/0000-0002-3094-4390. E-mail: timofeevEA@gmail.com
доктор. физ.-мат. наук., профессор кафедры теоретической информатики.

Для цитирования: E. A. Timofeev, “On a Segment Partition for Entropy Estimation”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 40-47, 2020.

© Тимофеев Е. А., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

В работе [1] для обоснования несмещенности оценки энтропии применялась последовательность разбиений отрезка, которая строилась по следующим рекуррентным правилам:

$$\begin{aligned} &Q_1 = \{0, q^2, q, 1\}. \\ &Q'_{n+1} = qQ_n \cap q^2Q_n, \quad Q''_{n+1} = q^2 + qQ_n \cap qQ_n, \quad Q'''_{n+1} = q^2 + qQ_n \cap q + q^2Q_n, \\ &Q_{n+1} = Q'_{n+1} \cup Q''_{n+1} \cup Q'''_{n+1}, \end{aligned} \tag{1}$$

где $q^2 + q = 1$.

В настоящей работе будет показано, что Q_n – измельчающая последовательность разбиений и отрезки разбиения Q_n имеют длины q^n, q^{n+1}, q^{n+2}. Найдено рекуррентное задание длин отрезков разбиения Q_n.

Введем последовательность чисел $d = 1, 2, 1, 0, 1, 2, 1, 0, 1, 0, 1, 2, 1, 0, 1, 2, 1, \dots$, положив

$$\begin{aligned} &d_1 = 1, \ d_2 = 2, \ d_4 = 0; \\ &d[2F_{2n} + 1 : 2F_{2n+1} + 1] = d[1 : 2F_{2n-1} + 1]; \\ &\quad n = 0, 1, 2, \dots; \\ &d[2F_{2n+1} + 2 : 2F_{2n+1} + 2F_{2n-2}] = d[2F_{2n-1} + 2 : 2F_{2n}]; \\ &d[2F_{2n+1} + 2F_{2n-2} + 1 : 2F_{2n+1} + 2F_{2n-1} + 1] = d[1 : 2F_{2n-3} + 1]; \\ &d[2F_{2n+1} + 2F_{2n-1} + 2 : 2F_{2n+2}] = d[2F_{2n-1} + 2 : 2F_{2n}]; \\ &\quad n = 1, 2, 3, \dots; \end{aligned} \tag{2}$$

где F_n – числа Фибоначчи ($F_{-1} = 0, F_0 = F_1 = 1$).

Теорема 1. *Пусть последовательность d задана в (2), тогда*

$$Q'_n = 1 - Q'''_n = \left\{ \sum_{i=1}^{k} q^{n+d_i}, \ k = 0, 1, \dots, m_n \right\}, \tag{3}$$

$$Q''_n = 1 - Q''_n = \left\{ q^2 + \sum_{i=m_n}^{k} q^{n+d_i}, \ k = m_n - 1, m_n, \dots, m_{n+1} \right\}, \tag{4}$$

где $m_{2n} = 2F_{2n-2}, \ m_{2n+1} = 2F_{2n-1} + 1$.

Замечание 1. *Последовательность d приведена и в «Энциклопедии целочисленных последовательностей»*[1] *под номером A191329, где она задана формулой*

$$d_n = [nq + n] \mod 2 + [nq] \mod 2.$$

Замечание 2. *Из (3) получаем формулу*

$$q^{2-n} = \sum_{i=1}^{m_n} q^{d_i}.$$

Доказательство. Для упрощения задания множеств Q_n докажем несколько вспомогательных лемм

[1] https://oeis.org/

Лемма 1. $Q'_n = 1 - Q'''_n$, $Q''_n = 1 - Q''_n$.

Доказательство. Проведем индукцию по n.

При $n = 1$ условия леммы выполнены.

Предположим, что они выполнены для n и покажем, что они выполнены при $n + 1$.

Из предположений следует, что $Q_n = 1 - Q_n$.

Из определения разбиения (1) следует, что

$$1 - Q'''_{n+1} = q - qQ_n \cap q^2 - q^2 Q_n = qQ_n \cap q^2 Q_n = Q'_{n+1}.$$

$$1 - Q''_{n+1} = q - qQ_n \cap 1 - qQ_n = q(1 - Q_n) \cap q^2 + q(1 - Q_n) = qQ_n \cap q^2 + qQ_n = Q''_{n+1}.$$

□

Лемма 2. *Справедливы соотношения*

$$Q'_n = qQ'_{n-1} \cup qQ''_{n-1}. \tag{5}$$

$$Q'_n \cap qQ'_n = qQ'_{n-1}, \tag{6}$$

$$Q'_n \cap qQ''_n = qQ''_{n-1}, \tag{7}$$

$$Q''_n \subset q^2 + qQ'_n, \tag{8}$$

$$Q''_n \subset qQ'''_n, \tag{9}$$

$$Q'_{n-1} \subset Q'_n, \tag{10}$$

$$Q''_{n-1} \subset Q''_n, \tag{11}$$

Доказательство. Проведем индукцию по n.

При $n = 2$ из (1) имеем

$$Q_2 = \{0, q^3, q^2, q, q + q^4, 1\},$$

поэтому условия леммы выполнены.

Предположим, что они выполнены для n и покажем, что они выполнены при $n + 1$.

Докажем (5). Из определения разбиения (1) следует, что

$$Q'_{n+1} = \left(qQ'_n \cup qQ''_n\right) \cap \left(q^2 Q'_n \cup q^2 Q''_n \cup q^2 Q'''_n\right) = q\left[\left(Q'_n \cap qQ'_n\right) \cup \left(Q'_n \cap qQ''_n\right) \cup \left(Q''_n \cap qQ'''_n\right)\right].$$

Применяя (6), (7), (9), получим

$$Q'_{n+1} = q\left(qQ'_{n-1} \cup qQ''_{n-1} \cup Q''_n\right).$$

Применяя (5), получим

$$Q'_{n+1} = q\left(Q'_n \cup Q''_n\right). \tag{12}$$

Докажем (6). Из последнего равенства (12) имеем

$$Q'_{n+1} \cap qQ'_{n+1} = \left(qQ'_n \cup qQ''_n\right) \cap \left(q^2 Q'_n \cup q^2 Q''_n\right) = q\left[\left(Q'_n \cap qQ'_n\right) \cup \left(Q'_n \cap qQ''_n\right)\right].$$

Применяя (6), (7), (5), получим

$$Q'_{n+1} \cap qQ'_{n+1} = q\left(qQ'_{n-1} \cup qQ''_{n-1}\right) = qQ''_n.$$

Докажем (7). Из определения разбиения (1) следует, что

$$Q''_{n+1} = q^2 + qQ'_n \cap qQ'''_n. \tag{13}$$

Отсюда и из (12) следует, что

$$Q'_{n+1} \cap qQ''_{n+1} = \left(qQ'_n \cup qQ''_n\right) \cap q^3 + q^2Q'_n \cap q^2Q'''_n = q\left[\left(Q'_n \cup Q''_n\right) \cap q^2 + qQ'_n \cap qQ'''_n\right] = \\ = q\left[Q''_n \cap q^2 + qQ'_n \cap qQ'''_n\right].$$

Применяя (8), (9), получим

$$Q'_{n+1} \cap qQ''_{n+1} = qQ''_n.$$

Докажем (8). Подставив (12) и (13), получим, что нужно доказать вложение

$$Q''_{n+1} = q^2 + qQ'_n \cap qQ'''_n \subset q^2 + q^2Q'_n \cup q^2 + q^2Q''_n.$$

Применяя (5), получим

$$q^2 + q\left(qQ'_{n-1} \cup qQ''_{n-1}\right) \cap qQ'''_n \subset q^2 + q^2Q'_n \cup q^2 + q^2Q''_n.$$

Последнее вложение выполняется по (10), (11).

Докажем (9). Из леммы 1 и (12) следует, что

$$Q'''_{n+1} = q^2 + qQ''_n \cap q^2 + qQ'''_n. \tag{14}$$

Подставив (13), получим, что нужно доказать вложение

$$Q''_{n+1} = q^2 + qQ'_n \cap qQ'''_n \subset q^3 + q^2Q''_n \cup q^3 + q^2Q'''_n = Q'''_{n+1}.$$

Для этого достаточно доказать, что

$$Q'''_n \subset q^2 + qQ''_n \cup q^2 + qQ'''_n.$$

Из леммы 1 и (5) следует, что

$$Q'''_n = q^2 + qQ''_{n-1} \cap q^2 + qQ'''_{n-1}.$$

Подставляя, получим вложение

$$q^2 + qQ''_{n-1} \cap q^2 + qQ'''_{n-1} \subset q^2 + qQ''_n \cup q^2 + qQ'''_n,$$

которое выполняется по (10), (11).

Докажем (10). Подставив (5), (12), получим, что нужно доказать вложение

$$Q'_n = qQ'_{n-1} \cup qQ''_{n-1} \subset qQ'_n \cup qQ''_n = Q'_{n+1},$$

которое выполняется по (10), (11).

Докажем (11). Подставив (13), получим, что нужно доказать вложение

$$Q''_n \subset Q''_{n+1} = q^2 + qQ'_n \cap qQ'''_n.$$

Применяя (10), (11), получим

$$Q''_{n+1} = q^2 + qQ'_n \cap qQ''_n \supset q^2 + qQ'_{n-1} \cap qQ'''_{n-1} = Q''_n.$$

□

Введем обозначения.

Пусть $d = d_1, d_2, \dots, d_m$ – последовательность целых чисел, тогда положим

$$\overline{d} = d_m, d_{m-1}, \dots, d_1, \tag{15}$$

$$d^* = 0, d_3, d_4, \dots, d_m, \tag{16}$$

$$Q(d) = \left\{ \sum_{i=1}^{k} q^{d_i},\ k = 0, 1, \dots, m \right\}. \tag{17}$$

Лемма 3.

$$Q'_n = q^n Q(d'_n), \tag{18}$$

$$Q''_n = q^2 + q^n Q(d''_n), \tag{19}$$

где последовательности d'_n, d''_n удовлетворяют рекуррентным уравнениям

$$d'_{2n} = d'_{2n-1} d''_{2n-1}, \quad d''_{2n} = d'_{2n-1}, \tag{20}$$

$$\begin{aligned} d'_{2n-1} &= d'_{2n-3} d''_{2n-3} d'_{2n-3}, \\ d'^*_{2n-1} &= d''_{2n-1} d'_{2n-3}, \\ d''_{2n-1} &= d''_{2n-3} d'_{2n-5} d''_{2n-3}. \end{aligned} \tag{21}$$

Доказательство. Проведем индукцию по n.

$$\begin{aligned} Q'_3 &= q^3 Q(d'_3) = \{0, q^4, q^3, q^2\},\ d'_3 = 1, 2, 1; \\ Q''_3 &= q^2 + q^3 Q(d''_3) = \{q^2, q\},\ d''_3 = 0; \end{aligned}$$

$$\begin{aligned} Q'_4 &= q^4 Q(d'_4) = \{0, q^5, q^4, q^3, q^2\},\ d'_4 = 1, 2, 1, 0; \\ Q''_4 &= q^2 + q^4 Q(d''_4) = \{q^2, q^2 + q^5, q^2 + q^4, q\},\ d''_4 = 1, 2, 1; \end{aligned}$$

$$\begin{aligned} Q'_5 &= q^5 Q(d'_5) = \{0, q^6, q^5, q^4, q^3, q^3 + q^6, q^3 + q^5, q^2\},\ d'_5 = 1, 2, 1, 0, 1, 2, 1; \\ Q''_5 &= q^2 + q^5 Q(d''_5) = \{q^2, q^2 + q^5, q^2 + q^4, q\},\ d''_5 = 0, 1, 0; \end{aligned}$$

Поэтому лемма справедлива при $n \le 5$. Предположим, что утверждения леммы выполняются при некотором n, и покажем, что они выполнены для $n + 1$.

Применяя (5) и индукционное предположение, получим

$$Q'_{n+1} = qQ'_n \cup qQ''_n = q^{n+1} Q(d'_n) \cup q^3 + q^{n+1} Q(d''_n) = q^{2n+1} Q(d'_n d''_n).$$

Следовательно,

$$d'_{n+1} = d'_n d''_n. \tag{22}$$

При нечетном n получаем первое равенство в (20). При четном n, подставляя (20), получим первое равенство в (21).

Из (13) имеем

$$Q''_{2n+1} = q^2 + q^{2n+2} Q(d'_{2n}) \cap q^2 + q^{2n+2} Q(\overline{d'}_{2n}) = q^2 + q^{2n+2} \left[Q(d'_{2n}) \cap Q(\overline{d'}_{2n}) \right].$$

Найдем общую часть последовательностей $d'_{2n} = d'_{2n-1} d''_{2n-1}$ и $\overline{d'}_{2n} = d''_{2n-1} \overline{d'}_{2n-1}$.

Из второго равенства в (21) имеем

$$d'^*_{2n-1} d''_{2n-1} = d''_{2n-1} d'_{2n-3} d''_{2n-1}.$$

Поэтому,

$$d'_{2n} = 1, 2, d''_{2n-1}d'_{2n-3}d''_{2n-1},\ \overline{d'}_{2n} = d''_{2n-1}d'_{2n-3}d''_{2n-1}, 2, 1.$$

Поскольку $q + q^2 = 1$,

$$Q(d'_{2n}) \cap Q(\overline{d'}_{2n}) = Q(d'^{*}_{2n-1}d''_{2n-1}).$$

Следовательно,

$$d''_{2n+1} = d''_{2n-1}d'_{2n-3}d''_{2n-1}$$

и третье равенство в (21) выполнено.

Подставляя полученное равенство в

$$d'^{*}_{2n+1} = d'^{*}_{2n}d''_{2n} = d'^{*}_{2n-1}d''_{2n-}d'_{2n-1} = d''_{2n-1}d''_{2n-3}d''_{2n-1}d'_{2n-1},$$

получим второе равенство в (21).

Докажем второе равенство в (20).

Из (13) имеем

$$Q''_{2n+2} = q^2 + q^{2n+3}Q(d'_{2n+1}) \cap q^2 + q^{2n+3}Q(\overline{d'}_{2n+1}) = q^2 + q^{2n+3}\left[Q(d'_{2n+1}) \cap Q(\overline{d'}_{2n+1})\right].$$

Поскольку

$$d'_{2n+1} = d'_{2n-1}d''_{2n-}d'_{2n-1} = \overline{d'}_{2n+1},$$

получаем

$$d''_{2n+2} = d'_{2n+1}.$$

□

Найдем величины $m_n = |d'_n|$.

Из (22) имеем

$$|d''_n| = m_{n+1} - m_n. \tag{23}$$

Подставляя во второе уравнение (21), получим

$$m_{2n} = m_{2n+1} - m_{2n-1}. \tag{24}$$

Подставляя (23) в первое уравнение (20), получим уравнение, равносильное (24).

Подставляя (23), (24) в третье уравнение (20), получим рекуррентное уравнение

$$m_{2n+1} - 4m_{2n-1} + 4m_{2n-3} - m_{2n-5} = 0 \tag{25}$$

с начальным условием $m_1 = 1$, $m_3 = 3$, $m_5 = 7$.

Корни характеристического уравнения равны 1, q^2, q^{-2}, их степени выражаются через числа Фибоначчи, поэтому общее решение уравнения (25) имеет вид

$$m_{2n+1} = C_0 + C_1F_{2n-1} + C_2F_{2n-2}.$$

Найдя константы из начальных условий, получим

$$m_{2n+1} = 2F_{2n-1} + 1.$$

Подставляя в(24), получим

$$m_{2n} = 2F_{2n-2}.$$

□

References

[1] E. Timofeev, “Existence of an unbiased consistent entropy estimator for the special Bernoulli measure”, *Modeling and Analysis of Information Systems*, vol. 26, no. 2, pp. 267–278, 2019.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

COMPUTING METHODOLOGIES AND APPLICATIONS

Modern Approaches to Detect and Classify Comment Toxicity Using Neural Networks

S. V. Morzhov[1]

DOI: 10.18255/1818-1015-2020-1-48-61

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 68T50
Research article
Full text in Russian

Received January 17, 2020
After revision February 25, 2020
Accepted February 28, 2020

The growth of popularity of online platforms which allow users to communicate with each other, share opinions about various events, and leave comments boosted the development of natural language processing algorithms. Tens of millions of messages per day are published by users of a particular social network need to be analyzed in real time for moderation in order to prevent the spread of various illegal or offensive information, threats and other types of toxic comments. Of course, such a large amount of information can be processed quite quickly only automatically. That is why there is a need to find a way to teach computers to "understand" a text written by humans. It is a non-trivial task even if the word "understand" here means only "to classify". The rapid evolution of machine learning technologies has led to ubiquitous implementation of new algorithms. A lot of tasks, which for many years were considered almost impossible to solve, are now quite successfully solved using deep learning technologies. This article considers algorithms built using deep learning technologies and neural networks which can successfully solve the problem of detection and classification of toxic comments. In addition, the article presents the results of the developed algorithms, as well as the results of the ensemble of all considered algorithms on a large training set collected and tagged by Google and Jigsaw.

Keywords: toxicity; Natural Language Processing; NLP; deep learning; word embedding; GloVe; FastText; recurrent neural networks; convolutional neural networks; CNN; LSTM; GRU

INFORMATION ABOUT THE AUTHORS

Sergey V. Morzhov | orcid.org/0000-0001-6652-3574. E-mail: smorzhov@gmail.com
postgraduate student.

For citation: S. V. Morzhov, "Modern Approaches to Detect and Classify Comment Toxicity Using Neural Networks", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 48-61, 2020.

© Morzhov S. V., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

COMPUTING METHODOLOGIES AND APPLICATIONS

Современные методы детектирования и классификации токсичных комментариев с использованием нейронных сетей

С. В. Моржов[1]

DOI: 10.18255/1818-1015-2020-1-48-61

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

УДК 004.8
Научная статья
Полный текст на русском языке

Получена 17 января 2020 г.
После доработки 25 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

Рост популярности онлайн-платформ, позволяющих пользователям общаться друг с другом, делиться мнениями о различных событиях, оставлять комментарии, подтолкнул к развитию алгоритмов обработки естественного языка. Десятки миллионов сообщений в день, которые публикуют пользователи отдельно взятой социальной сети, необходимо анализировать в режиме реального времени или близко к тому с целью модерации, чтобы не допустить распространение различной противозаконной или оскорбительной информации, угроз и других видов токсичных комментариев. Разумеется такой большой объем информации может быть обработан достаточно быстро только автоматически. Возникает необходимость научить компьютер «понимать» текст, написанный человеком, что является нетривиальной задачей, пусть даже под «пониманием» текста подразумевается лишь его классификация. Бурное развитие технологий машинного обучения обусловило повсеместное внедрение новых алгоритмов. Многие задачи, в том числе и задачи обработки естественного языка, которые долгие годы считалось практически невозможно решить, сейчас вполне успешно решаются с использованием технологий глубокого обучения. В данной статье будут рассмотрены алгоритмы, построенные с использованием технологий глубокого обучения и нейронных сетей, позволяющие успешно решать задачу распознавания и классификации токсичных комментариев. Помимо этого, в статье будут приведены результаты тестирования как разработанных алгоритмов, так и ансамбля данных алгоритмов на большой обучающей выборке, собранной и размеченной специалистами компаний Google и Jigsaw.

Ключевые слова: токчисность; обработка естественного языка; NLP; глубокое обучение; векторное представление слов; GloVe; FastText; реккурентные нейронные сети; сверточные нейронные сети; CNN; LSTM; GRU

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сергей Владимирович Моржов | orcid.org/0000-0001-6652-3574. E-mail: smorzhov@gmail.com
аспирант.

Для цитирования: S. V. Morzhov, “Modern Approaches to Detect and Classify Comment Toxicity Using Neural Networks”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 48-61, 2020.

© Моржов С. В., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

Задача обработки естественного языка уже многие годы является привлекательной целью исследований, так как решение ее в общем виде позволит создать естественно-языковой интерфейс, что существенно упростит и расширит сферы взаимодействия человека с компьютером. Само по себе понимание естественного языка — нетривиальная задача, считающаяся AI-полной, потому как распознавание естественного языка требует больших знаний об окружающем мире и возможности о взаимодействия с ним. Однако, при решении отдельных классов задач, например, классификации или анализа тональности текста, в последнее время был достигнут большой прогресс благодаря развитию нейросетевых алгоритмов, а также появлению высокопроизводительных процессоров и графических карт. Это позволило использовать глубокие нейронные сети для решения различных задач, связанных с обработкой естественного языка, которые ранее не могли быть успешно решены с применением классических алгоритмов.

В последнее время широкое распространение получили онлайн-платформы, позволяющие пользователям различные виды взаимодействия друг с другом, в том числе посредством обмена сообщениями. Различные социальные сети, платформы онлайн-игр, приложения для обмена фотографиями и видео, новостные порталы встраивают в свои продукты чаты, реализуют возможность оставлять комментарии, позволяют пользователям общаться друг с другом. Данный функционал уязвим перед многими видами Интернет-преступлений, среди которых можно выделить: личные оскорбления и угрозы, различные виды пропаганды, мошенничество, реклама незаконных товаров и услуг. Противоправные и токсичные комментарии должны удаляться, хотя, бесспорно, лучшим вариантом будет служить возможность запрета их публикации. Это предусматривает наличие достаточно быстрых и эффективных алгоритмов, способных в режиме реального времени обрабатывать все сообщения пользователей.

Компанией Jigsaw, занимающейся проблемами безопасности в Интернете, совместно с Google проводился конкурс [1], целью которого было создание алгоритма, решающего задачу детектирования токсичных комментариев. Это говорит об актуальности, а также низком уровне исследований данной проблемы, так как опубликованные алгоритмы, позволяющие решать поставленную задачу (см. [2—5]), имеют недостаточную по мнению организаторов конкурса аккуратность предсказаний.

В данной статье будут представлены новые алгоритмы, позволяющие решать задачу детектирования и классификации токсичных комментариев. Также будут приведены сравнительные результаты тестирования разработанных алгоритмов и некоторых существующих алгоритмов, решающих эту задачу. Помимо этого, в статье содержится ряд замечаний касательно проведения дальнейшей работы по улучшению качества предсказаний представленных алгоритмов.

1. Анализ обучающей выборки

Для решения поставленной задачи использовалась обучающая выборка, подготовленная и размеченная специалистами компаний Google и Jigsaw. Данная выборка составлена из комментариев со страниц обсуждения статей Википедии. Размер предоставленной тренировочной части выборки составляет примерно 160 тысяч комментариев. Также предоставлена тестовая выборка, содержащая примерно 154 тысячи комментариев.

Обучающая выборка размечена следующим образом. Каждому комментарию соответствует шесть меток: токсичность (toxic), сильная токсичность (severe toxic), непристойность (obscene), угроза (threat), оскорбление (insult), ненависть к личности (identity hate). Метки принимают значение 1, если в комментарии есть данный тип токсичности, 0 — иначе. Случай, когда все метки нулевые означает, что комментарий не токсичен. Может быть так, что один комментарий содержит несколько типов токсичности. Разметка данных проводилась с помощью краудсорсинга (метки выставляли разные люди), а значит возможно наличие различных ошибок.

Тренировочная и тестовая выборки не содержат пропущенных значений. Тренировочная выборка крайне не сбалансирована — есть большой перекос в сторону «чистых», неоскорбительных, комментариев, что логично, так как в реальной жизни таких комментариев тоже обычно больше. Количество меток для каждого класса представлено на рисунке 1. Длины комментариев также сильно различаются (см. рисунок 2) и имеют следующие характеристики: минимальная длина — 2 символа, максимальная — 5000 символов, математическое ожидание — 394.07, среднеквадратическое отклонение — 590.72.

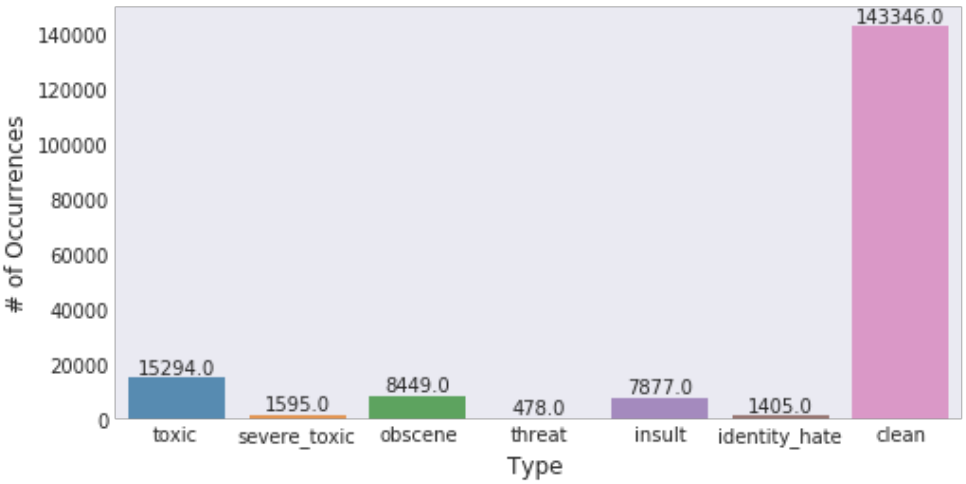


Fig. 1. Number of labels of each class

Рис. 1. Количество меток каждого класса

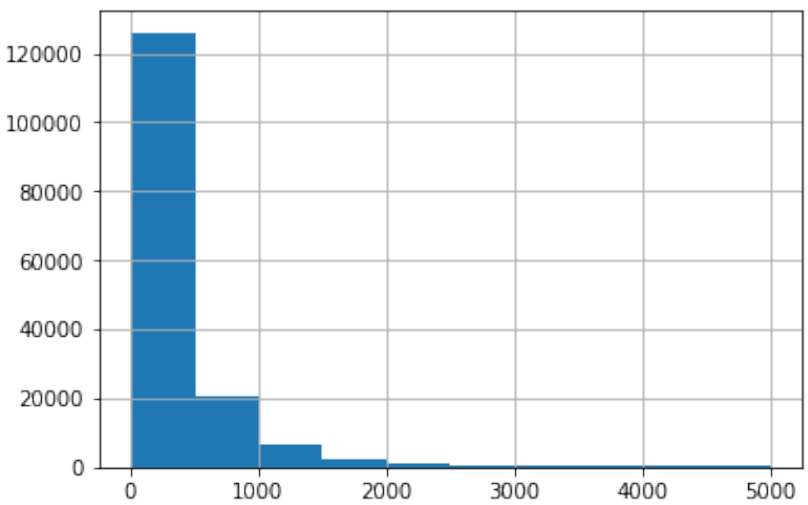


Fig. 2. Comment length distribution

Рис. 2. Распределение длины комментариев

Как уже было сказано выше, комментарий может иметь несколько меток, равных 1. Количество комментариев в зависимости от количества меток, равных 1, приведено на рисунке 3. Помимо этого, стоит также рассмотреть корреляцию между метками, равными 1 (см. рисунок 4). Исходя из анализа матрицы корреляции, можно сделать вывод, что класс «toxic» сильно коррелирует с классами «obscene» и «insult» (0.68 и 0.65), класс «insult» и «obscene» также имеют высокий индекс

корреляции (0.74). Интересным является индекс корреляции между классами «toxic» и «severe toxic», оказавшийся достаточно низким (0.31). В силу того, что количество комментариев, принадлежащих к этим классам сильно различается, стоит воспользоваться другим методом оценки корреляции.

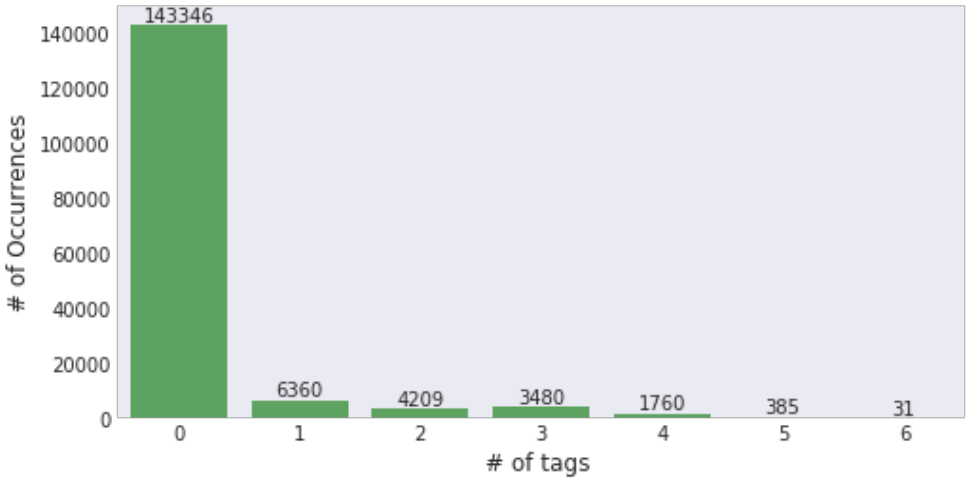


Fig. 3. Number of comments depending on the number of tags

Рис. 3. Количество комментариев в зависимости от количества меток

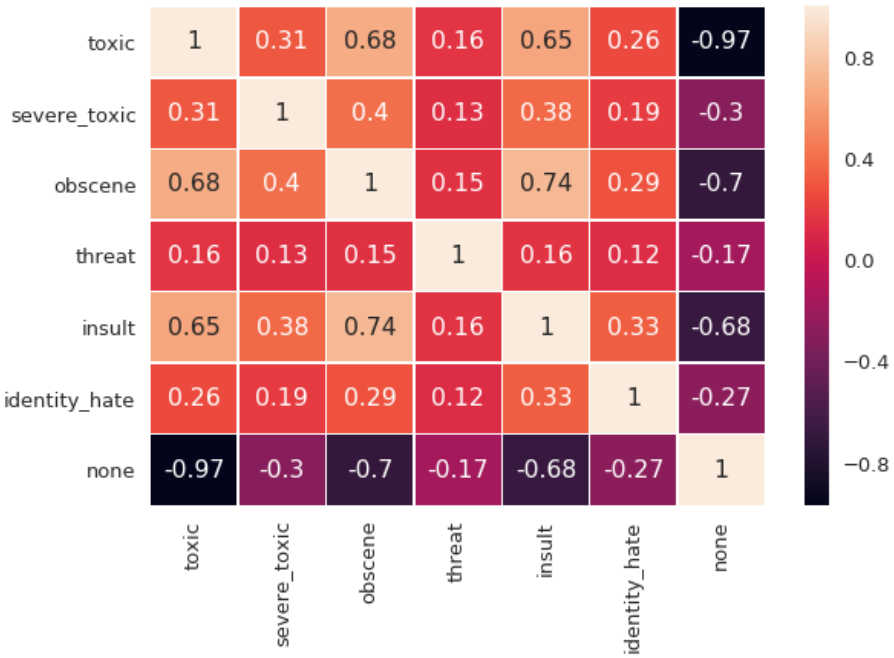


Fig. 4. Toxic labels correlation matrix

Рис. 4. Матрица корреляции меток

Рассмотрим таблицу сопряженности для метки «toxic» и всех остальных меток (см. таблицу 1). Можно отметить, что комментарии, отмеченные как «severe toxic», всегда будет также отмечены как «toxic». Другие классы, похоже, также являются подмножествами класса «toxic» за некоторыми исключениями.

Table 1. Cross tabulation for the “toxic” tag

Таблица 1. Таблица сопряженности для метки «toxic»

	severe toxic		obscene		threat		insult		identity hate	
severe toxic	0	1	0	1	0	1	0	1	0	1
toxic										
0	144277	0	143754	523	144248	29	143744	533	144174	103
1	13699	1595	7368	7926	14845	449	7950	7344	13992	1302

Существует еще одна важная особенность, на которую необходимо обратить внимание: количество уникальных слов в каждом комментарии. Данная характеристика может помочь при решении поставленной задачи, так как нетрудно заметить, что авторы токсичных комментариев не очень изобретательны в своей лексике. Иными словами, необходимо проверить следующую гипотезу: есть ли корреляция между различными характеристиками комментариев, связанными с количеством уникальных слов и токсичностью комментария. Нетрудно отметить, что существуют заметные сдвиги в среднем числе слов и количестве уникальных слов в чистых и токсичных комментариях (см. рисунок 5). Кроме этого, если рассмотреть график на рисунке 6, то можно заметить, что рядом с отметкой 0 – 10% имеется выпуклость, указывающая на большое количество токсичных комментариев, которые содержат небольшое количество разнообразных слов.

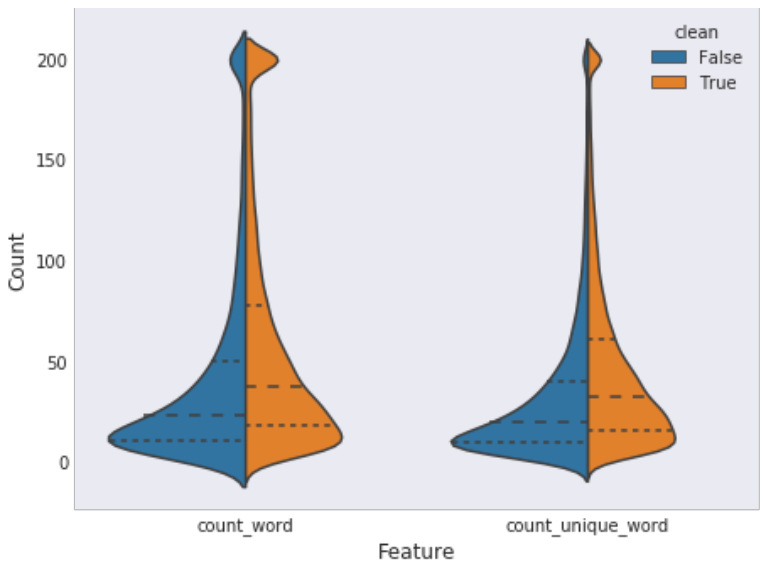


Fig. 5. Absolute number of words and number of unique words

Рис. 5. Абсолютное количество слов и количество уникальных слов

Все эти наблюдения необходимо учитывать при построении нейросетевых алгоритмов для решения задачи детектирования и классификации токсичных комментариев. Данные статистические особенности обучающей выборки помогут в дальнейшем при интерпретации результатов, а также их можно использовать для отладки и улучшения аккуратности предсказаний разработанных моделей.

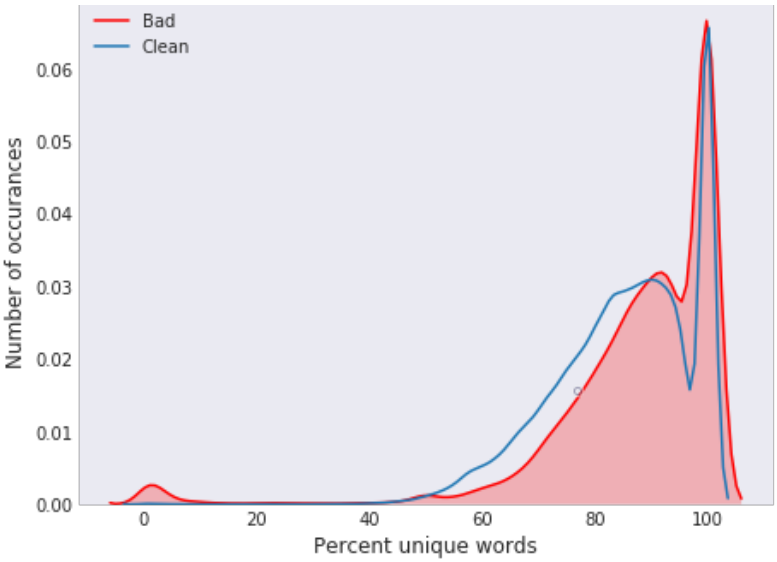


Fig. 6. Percentage of unique words in the total number of words in a comment

Рис. 6. Процент уникальных слов от общего количества слов в комментарии

Помимо статистических особенностей обучающей выборки необходимо также рассмотреть особенности текстовых данных. Комментарии преимущественно написаны на английском языке (в выборке присутствуют комментарии на других языках, но их количество составляет менее 0.1% от общего числа). Многие комментарии содержат эмодзи, избыточное количество пунктуационных знаков, веб-ссылки, числа, особые начертания слов, грамматические ошибки. Некоторые грамматические ошибки были допущены авторами специально, чтобы «замаскировать» оскорбительные или нецензурные высказывания. Все это излишне расширяет размер словаря, что в свою очередь усложняет анализ комментариев, поэтому подготовка данных из обучающей и тренировочной выборки является важным этапом при решении поставленной задачи.

2. Подготовка данных

Для решения задачи использовалась двухэтапная подготовка данных. На первом этапе осуществлялись базовые, наиболее простые манипуляции с ними:

1. Приведение текста к нижнему регистру.
2. Удаление кусков html-кода, которые присутствуют в некоторых комментариях.
3. Преобразование подстроки вида «w h a t a n i c e d a y» к «what a nice day». Данный вид искажения текста часто используется для маскировки нецензурных слов.
4. Удаление ссылок, ip-адресов.
5. Удаление чисел и цифр.
6. Удаление всей пунктуации, кроме «’», «.», «!», «?». При этом также удалялись дублирующиеся знаки, то есть подстрока вида «!!!!!» приводилась к «!».
7. Замена знаков окончания предложения на специальные токены. «!» заменялся на « exclmrk », «?» — на « qstmrk », «.» — на « eosmkr ». Это было сделано для того, чтобы не потерять информацию об этих знаках на этапе трансформации текста в векторное представление.

На втором этапе проводились более трудоемкие операции исправления и очистки текста.

1. Замена эмодзи на соответствующие слова («:-(» на «sad» и т.д.).
2. Расшифровка некоторых сокращений («won't» на «will not», «'ll» на « will» и т.д.).
3. Исправление нецензурных слов. «f*ck», «fu**» и т.п. заменялись соответствующими словами без звездочек.
4. Исправление иных грамматических ошибок [1].

Обработку данных было решено разбить на две части также для того, чтобы уменьшить взаимную корреляцию моделей на этапе ансамбля. В процессе обучения каждая модель обучалась на данных, прошедших либо только первый этап обработки, либо оба этапа, что повышало вариативность предсказаний.

3. Модели

Для решения задач анализа текста хорошо себя зарекомендовали рекуррентные нейронные сети (РНС). Рекуррентные нейронные сети (РНС) используют идею обработки последовательной информации. Термин «рекуррентные» используется для того, чтобы показать, что нейронные сети выполняют одну и ту же задачу для каждого экземпляра последовательности, так что выходные данные зависят от предыдущих вычислений и результатов. Как правило, вектор фиксированного размера создается для представления последовательности путем подачи лексем одной за другой в рекуррентный блок. В некотором смысле, РНС «запоминает» предыдущие вычисления и используют эту информацию в текущей обработке. С задачами классификации текста лучше всего справляются такие разновидности РНС, как LSTM [6] и GRU [7], поэтому при построении моделей для решения поставленной задачи детектирования и классификации токсичных комментариев было решено использовать именно их.

3.1. Векторное представление слов

При решении любой задачи NLP неизбежно встает проблема представления слов в понятном для компьютера виде. Наиболее тривиально данную задачу можно решить следующим образом: закодируем все слова словаря цифрами по порядку. Так как натуральный ряд бесконечен, то не составит труда перенумеровать все слова. Однако, у этой идеи есть существенный недостаток. Согласно гипотезе лингвиста Фердинанда де Соссюра, буквенное написание слова совершенно не связано с его смыслом. Возьмем, к примеру, слова «петух», «курица», «цыпленок». В словаре они находятся далеко друг от друга, хотя очевидно обозначают самца, самку и детеныша одного вида птиц. Таким образом, можно выделить два вида близости слов: лексическую и семантическую. Пример с курицей показывает, что они не всегда совпадают. Для наглядности можно привести обратный пример лексически близких, но семантически далеких слов: «зола» и «золото».

Существует гипотеза, согласно которой слова, имеющие близкое значение, как правило употребляются в текстах в близких по смыслу контекстах. Отсюда вытекает, что семантическая близость слов является важным признаком, который необходимо сохранить при представлении слов в понятном для компьютера виде.

Одним из способов решения поставленной задачи является сопоставление словам (и, возможно, фразам) из некоторого конечного фиксированного словаря размера N векторов из $\mathbb{R}^n$, $n << N$. Данная техника называется векторное представление слов (англ. word embedding). Так как векторы должны отражать семантическую близость слов, то пусть близкие векторы (например, по косинусной мере), обозначают близкие по смыслу слова.

[1] Данный вид преобразования в конечном итоге решено было не использовать из-за слишком длительной обработки в силу размера корпуса.

Векторное представление слов часто используется в качестве первого слоя, преобразующего текст на естественном языке в вид, пригодный для дальнейшей обработки какой-либо моделью глубокого обучения. Для данной задачи были выбраны две модели генерации векторного представления слов: GloVe [8] и FastText [9].

3.2. Bi-GRU-LSTM

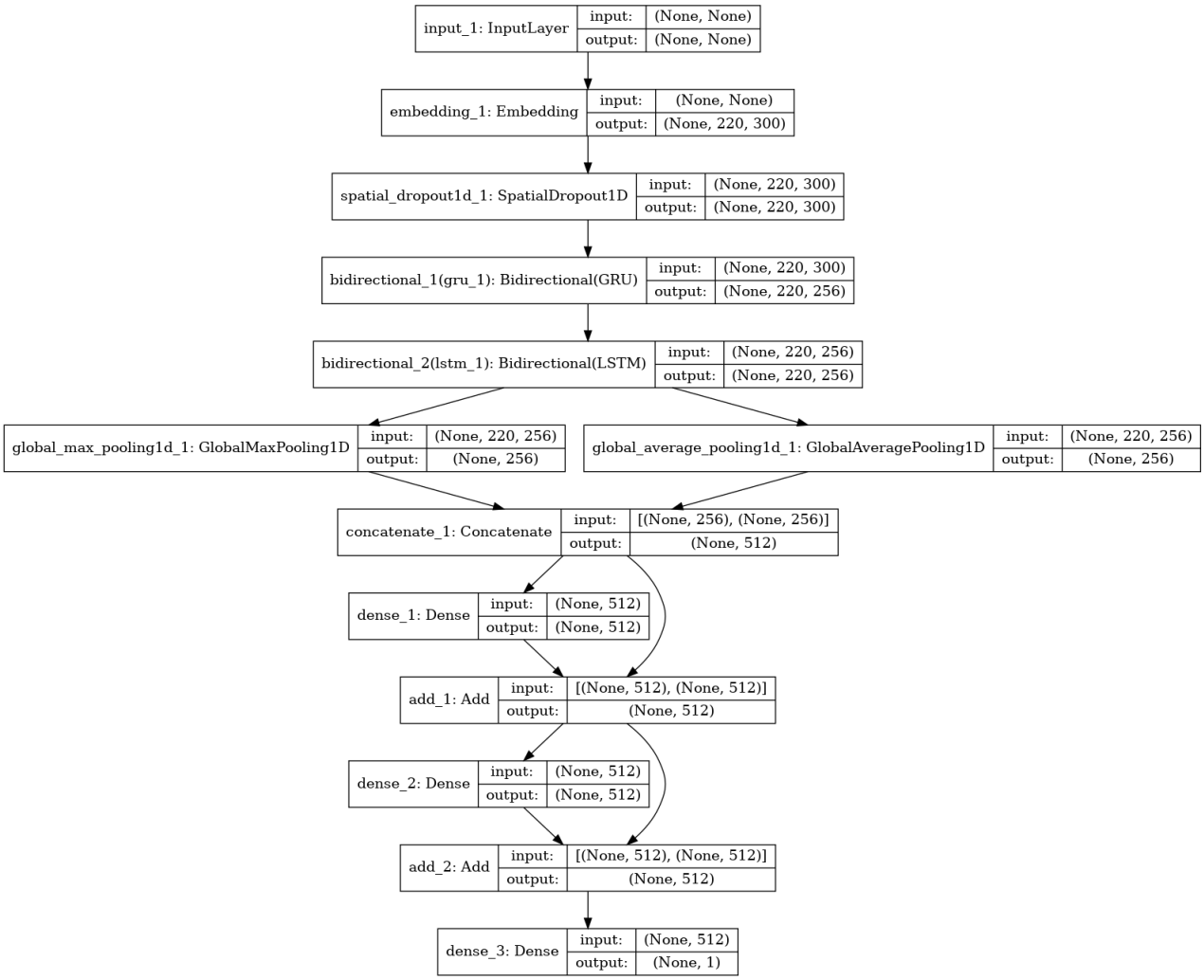


Fig. 7. The architecture of Bi-GRU-LSTM model

Рис. 7. Архитектура модели Bi-GRU-LSTM

Модель, сочетающая в себе стекинг GRU и LSTM слоев, была выбрана в качестве базовой для решения задачи классификации токсичных комментариев [10]. Она достаточно глубока, чтобы выделить необходимое количество признаков из не очень большой тренировочной выборки (всего порядка 140000 комментариев, принимая во внимание использование кросс-валидации на 10 частях). Архитектура модели Bi-GRU-LSTM представлена на рисунке 7.

3.3. Bi-GRU с механизмом внимания

Данная модель похожа на Bi-GRU-LSTM, за тем исключением, что здесь последовательно используются два слоя GRU и механизм внимания [11]. В оригинальной статье механизм внимания использовался на примере задачи машинного перевода. Однако ничего не мешает применить его

и к задаче классификации [12]. По своей сути механизм внимания — это не что иное, как усовершенствованный механизм кодеров-декодеров. Использование GRU или LSTM позволяет сохранить информацию о структуре последовательности, при этом не позволяя назначать различные веса элементам полученных последовательностей. Очевидно, что в задаче классификации токсичных комментариев различные слова будут иметь разную значимость: нецензурные слова должны иметь больший вес, являясь хорошим сигналом, что комментарий вероятно токсичный. Ни LSTM, ни GRU не способны моделировать подобное в отличие от механизма внимания. Иными словами, механизм внимания позволяет присвоить различные веса каждому элементу обрабатываемой последовательности. Чем вес больше, тем важнее данное слово, следовательно, данному слову нужно уделить больше внимания.

Архитектура модели Bi-GRU-LSTM представлена в Приложении A на странице 59.

3.4. Асимметричный CNN-LSTM

Данная модель является усложнением модели Bi-GRU с механизмом внимания. Были добавлены несколько сверточных слоев [13] для выделения признаков перед первым рекуррентным блоком.

Архитектура данной модели представлена в Приложении B на странице 60.

4. Результаты тестирования

В качестве метрики оценки аккуратности предсказаний разработанных моделей было решено использовать усредненный по типам токсичности ROC AUC, то есть среднее арифметическое ROC AUC (Avg. ROC AUC) для каждого класса в отдельности.

В таблице 2 приведены результаты работы различных моделей на подготовленных данных, полученных согласно описанному ранее алгоритму очистки (использовалась двухэтапная подготовка). Во второй колонке содержится среднее арифметическое Avg. ROC AUC по всем 10 фолдам.

Table 2. Comparison of the results of different toxic comments classification models

Таблица 2. Сравнение результатов работы различных моделей для классификации токсичности комментариев

Модель	10-fold cross-validation Avg. ROC AUC	Test Avg. ROC AUC
CNN [2]	0.9236	0.9120
LSTM [3]	0.9545	0.9492
LSTM-CNN [4]	0.9790	0.9778
Bi-GRU-LSTM	0.9887	0.9849
Bi-GRU с механизмом внимания	0.9888	0.9848
Асимметричный CNN-LSTM	0.9910	0.9836
Ансамбль	0.9889	0.9870

В ансамбль были включены все модели, представленные в данной статье, с векторным представлением слов GloVe и FastText. Таким образом, в ансамбль были включены 6 моделей. Помимо этого использовалась техника, называемая «seed averaging», заключающаяся в запуске одной модели с инициализацией генератора псевдослучайных чисел разными значениями. Полученные таким образом предсказания усреднялись.

5. Заключение

Технологии глубокого обучения позволяют минимизировать участие человека в разработке алгоритмов, поскольку создание признаков, свойственных конкретной задаче, автоматизируется. В данной статье было показано, как с их помощью можно решить задачу детектирования и классификации токсичных комментариев. Полученные результаты как ансамбля, так и каждой разработанной модели в отдельности превзошли результаты из ранее опубликованных работ по данной тематике, что говорит об успешности проведенных исследований.

Дальнейшее совершенствование аккуратности предсказания моделей может быть достигнуто при использовании техники аугментации, т.е. увеличения размера тренировочной выборки. Это можно сделать, например, используя технику машинного перевода. Перед каждой эпохой во время тренировки модели можно выбирать некоторое подмножество комментариев из обучающей выборки, перевести их на какой-нибудь язык, например, немецкий, затем перевести полученные комментарии обратно на английский. Такое преобразование не должно сильно исказить смысл комментариев, но при этом увеличит размер выборки. Существует также более простой вариант увеличения тренировочной выборки. Он состоит в следующем: на этапе обучения перед каждой эпохой можно выбрать некоторое подмножество комментариев из обучающей выборки, часть из которых будет склеена друг с другом. Таким образом, будут получены новые, более длинные комментарии. Метки для них можно назначить путем объединения множества меток исходных комментариев.

Также можно попробовать совместить технологии глубокого обучения с деревьями решений и градиентным бустингом [14]. В ходе подготовки и очистки данных некоторая часть информации была потеряна. Ее можно восстановить вручную, создав новые признаки, вектор из которых, совместно с вектором предсказаний разработанных моделей, следует использовать для поиска более оптимальных предсказаний. К таким новым признакам можно отнести следующие: количество знаков восклицания (токсичные комментарии обычно содержат много знаков восклицания), количество вопросительных знаков, количество грамматических ошибок, длина комментария, количество нецензурных слов и т.п. Подобное усложнение модели не должно серьезно сказаться на времени обучения и трудоемкости при использовании, но согласно проведенным грубым оценкам, способно повысить качество предсказаний примерно на 0.001 – 0.002 по выбранной метрике Avg. ROC AUC.

Приложение A Bi-GRU с механизмом внимания

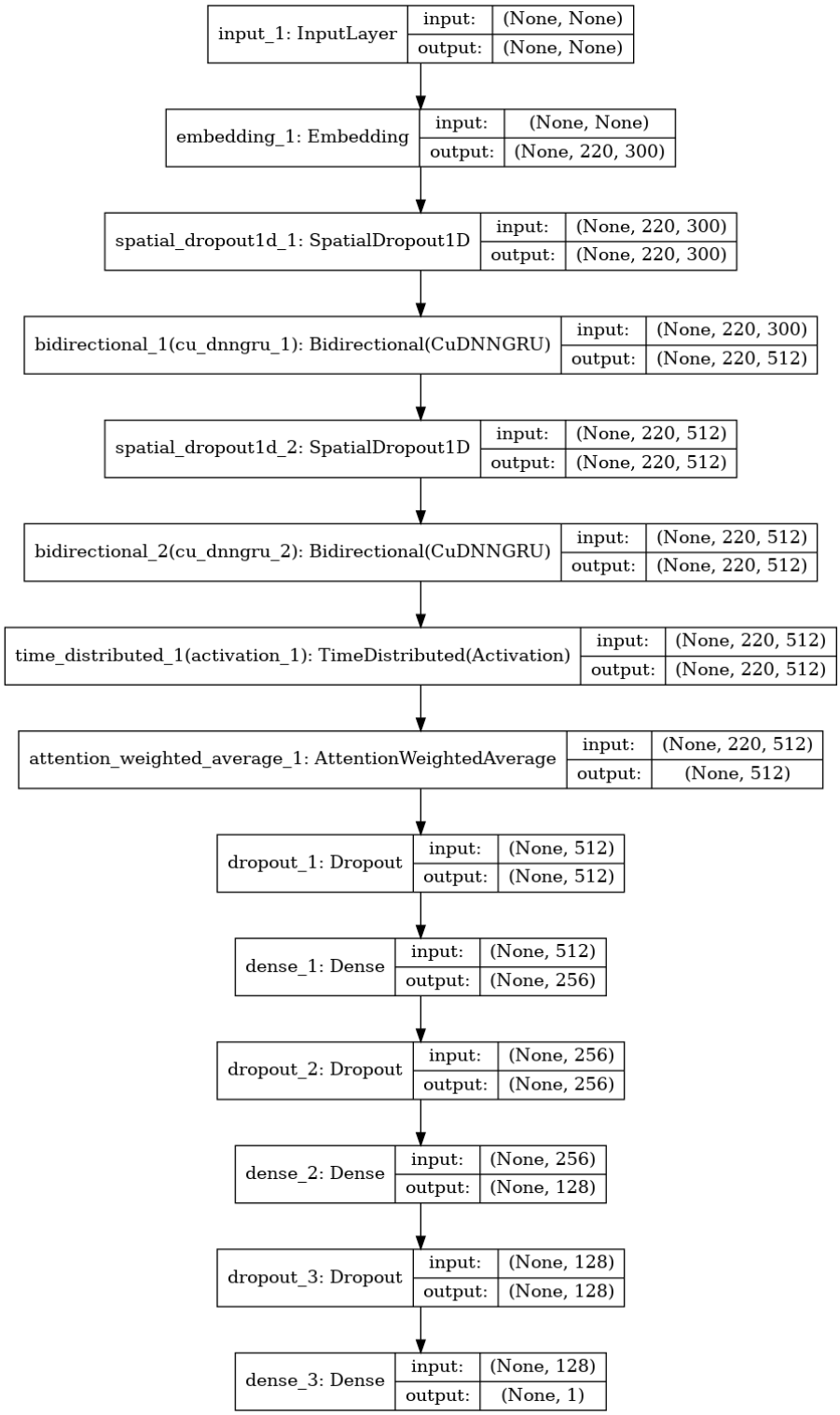


Fig. 8. The architecture of Bi-GRU with attention mechanism model

Рис. 8. Архитектура модели Bi-GRU с механизмом внимания

Приложение В Асимметричный CNN-LSTM

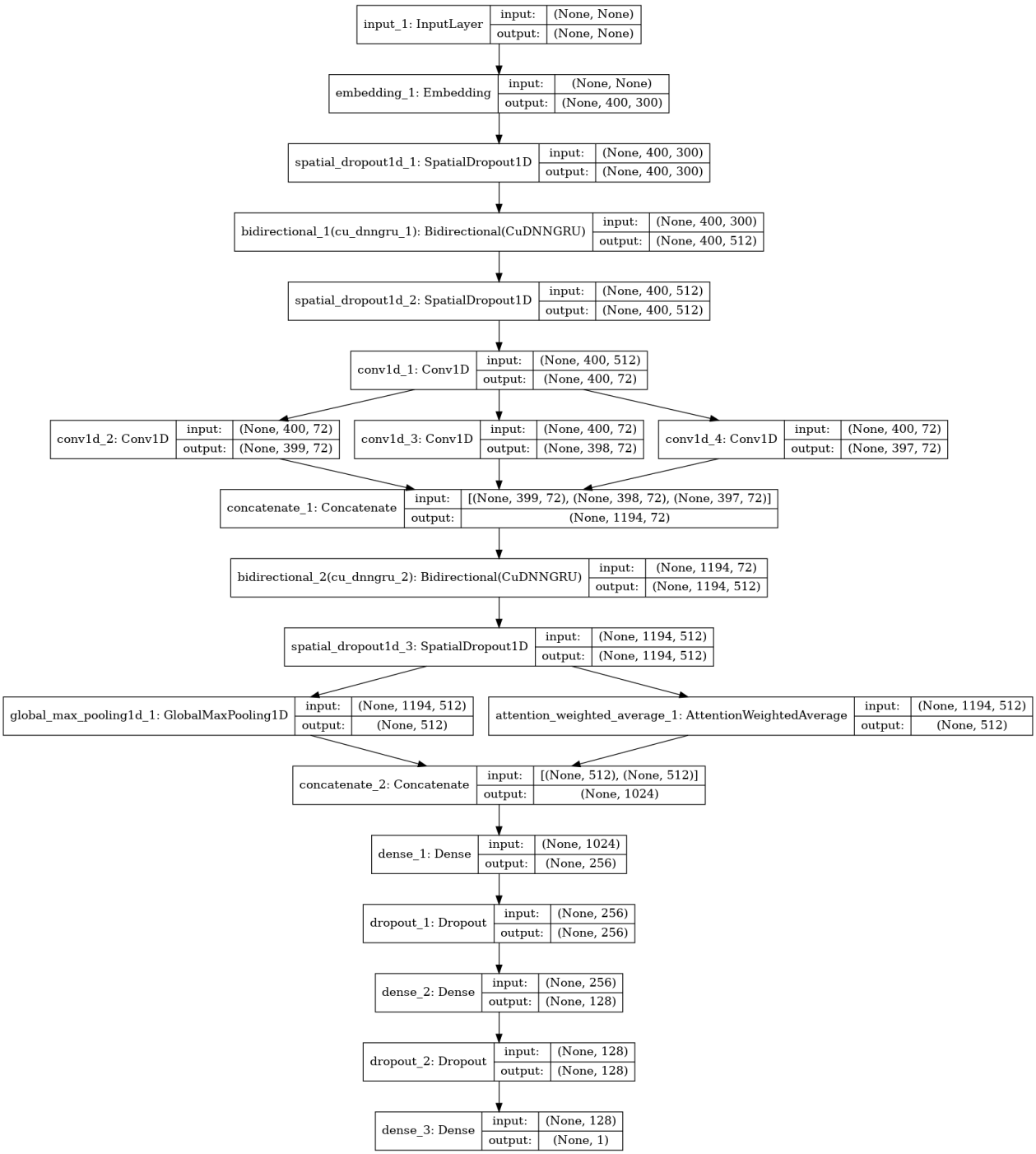


Fig. 9. The architecture of asymmetric CNN-LSTM model

Рис. 9. Архитектура асимметричной CNN-LSTM модели

References

[1] *Toxic Comment Classification Challenge*. [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview.

[2] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification", in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–6. arXiv: https://arxiv.org/pdf/1802.09957.pdf.

[3] M. Kohli, E. Kuehler, and J. Palowitch, *Paying attention to toxic comments online*. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6856482.pdf.

[4] T. Chu, J. K., and M. Wang, *Comment Abuse Classification with Deep Learning*. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762092.pdf.

[5] K. Khieu and N. N., *Detecting and Classifying Toxic Comments*. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

[8] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation", in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[9] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification", *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2, pp. 427–431, 2017.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", *arXiv preprint arXiv:1412.3555*, 2014.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

[12] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification", in *Proceedings of NAACL-HLT*, 2016, pp. 1480–1489. [Online]. Available: https://www.cs.cmu.edu/%5C%20./hovy/papers/16HLT-hierarchical-attention-networks.pdf.

[13] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical text classification using convolutional neural networks", *Stud Health Technol Inform*, vol. 235, pp. 246–50, 2017.

[14] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey", *Information*, vol. 10, no. 4, p. 150, 2019.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

COMPUTING METHODOLOGIES AND APPLICATIONS

Hierarchical Clustering as a Dimension Reduction Technique for Markowitz Portfolio Optimization

A. Y. Poletaev[1], E. M. Spiridonova[1]

DOI: 10.18255/1818-1015-2020-1-62-71

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 62-08
Research article
Full text in Russian

Received December 12, 2019
After revision February 14, 2020
Accepted February 28, 2020

Optimal portfolio selection is a common and important application of an optimization problem. Practical applications of an existing optimal portfolio selection methods is often difficult due to high data dimensionality (as a consequence of the large number of securities available for investment). In this paper, a method of dimension reduction based on hierarchical clustering is proposed. Clustering is widely used in computer science, a lot of algorithms and computational methods have been developed for it. As a measure of securities proximity for hierarchical clustering Pearson pair correlation coefficient is used. Further, the proposed method's influence on the quality of the optimal solution is investigated on several examples of optimal portfolio selection according to the Markowitz Model. The influence of hierarchical clustering parameters (intercluster distance metrics and clustering threshold) on the quality of the obtained optimal solution is also investigated. The dependence between the target return of the portfolio and the possibility of reducing the dimension using the proposed method is investigated too. For each considered example in the paper graphs and tables with the main results of the proposed method - application which are the decrease of the dimension and the drop of the yield (the decrease of the quality of the optimal solution) - for a portfolio constructed using the proposed method compared to a portfolio constructed without the proposed method are given. For the experiments the Python programming language and its libraries: scipy for clustering and cvxpy for solving the optimization problem (building an optimal portfolio) are used.

Keywords: clustering; optimization; Markowitz portfolio

INFORMATION ABOUT THE AUTHORS

Anatoliy Y. Poletaev correspondence author	orcid.org/0000-0003-0116-4739. E-mail: anatoliy-poletaev@mail.ru graduate student.
Elena M. Spiridonova	orcid.org/0000-0002-1089-7072. E-mail: lena@uniyar.ac.ru Sc.D.

For citation: A. Y. Poletaev and E. M. Spiridonova, "Hierarchical Clustering as a Dimension Reduction Technique for Markowitz Portfolio Optimization", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 62-71, 2020.

© Poletaev A. Y., Spiridonova E. M., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

Иерархическая кластеризация как метод снижения размерности в задаче оптимизации инвестиционного портфеля Марковица

А. Ю. Полетаев[1], Е. М. Спиридонова[1]

DOI: 10.18255/1818-1015-2020-1-62-71

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003, Россия.

УДК 311.2:004.021
Научная статья
Полный текст на русском языке

Получена 12 декабря 2019 г.
После доработки 14 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

Составление оптимального портфеля ценных бумаг является важным и частым случаем решения задачи оптимизации. Практическое применение существующих методов составления оптимального портфеля часто затруднено из-за большого числа доступных для инвестирования ценных бумаг (и, как следствие, большой размерности исходных данных). В данной работе предлагается метод снижения размерности исходных данных, основанный на иерархической кластеризации доступных для инвестирования ценных бумаг. Для кластеризации, широко используемой в компьютерных науках, уже разработано множество алгоритмов и методов. В качестве меры близости ценных бумаг для иерархической кластеризации используется коэффициент парной корреляции Пирсона. Далее исследуется влияние предложенного метода на качество получаемого оптимального решения на нескольких примерах составления оптимального портфеля ценных бумаг по модели Марковица. Также исследуется влияние параметров иерархической кластеризации (метрики межкластерного расстояния и порогового значения кластеризации) на изменение качества получаемого оптимального решения. Исследуется зависимость между целевой доходностью портфеля и возможностью снижения размерности с помощью предложенного метода. Для каждого рассмотренного примера приводятся графики и таблицы с основными полученными результатами применения метода — понижением размерности и падением доходности (снижением качества оптимального решения) у портфеля, построенного с применением предложенного метода по сравнению с портфелем, построенным без применения предложенного метода. Для проведения экспериментов используется язык программирования Python и его библиотеки: scipy для проведения кластеризации и cvxpy для решения задачи оптимизации (построения оптимального портфеля).

Ключевые слова: кластеризация; оптимизация; портфель Марковица

ИНФОРМАЦИЯ ОБ АВТОРАХ

Анатолий Юрьевич Полетаев
автор для корреспонденции
orcid.org/0000-0003-0116-4739. E-mail: anatoliy-poletaev@mail.ru
магистрант.

Елена Михайловна Спиридонова
orcid.org/0000-0002-1089-7072. E-mail: lena@uniyar.ac.ru
докт. экон. наук, доцент.

Для цитирования: A. Y. Poletaev and E. M. Spiridonova, “Hierarchical Clustering as a Dimension Reduction Technique for Markowitz Portfolio Optimization”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 62-71, 2020.

© Полетаев А. Ю., Спиридонова Е. М., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

Составление оптимального портфеля ценных бумаг является важным и частым случаем решения задачи оптимизации. Согласно портфельной теории, впервые сформулированной Гарри Марковицем в 1952 г., для составления оптимального портфеля из n ценных бумаг необходимо оценить лишь два показателя [1].

1. ожидаемую доходность $R = \sum_{i=1}^{n} R_i X_i$;
2. меру риска (изменчивости) $V = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{i,j} X_i X_j$.

Здесь R_i — ожидаемая доходность i-ой ценной бумаги; X_i — доля средств, инвестированных в неё ($\sum_{i=1}^{n} X_i = 1$); $\sigma_{i,j}$ — ковариация доходностей ценных бумаг i и j.

Портфель может быть оптимизирован по заданной ожидаемой доходности (для минимизации риска), по заданному риску (для максимизации доходности) и по RAPOC (risk-adjusted return) — тогда максимизируется функция $R - \gamma V$, где γ — некоторый коэффициент. Результатом оптимизации является вектор долей $X = [X_1, \dots, X_n]$.

В настоящее время разработано достаточно много математических методов оптимизации портфеля по Марковицу [2, 3], однако их общим недостатком является достаточно высокая вычислительная сложность. Учитывая, что объём биржевых данных, как правило, велик (например, в 2015 году только на Нью-Йоркской фондовой бирже торговались акции более 3000 компаний), а оптимизация портфеля на динамичном рынке может требоваться достаточно часто, необходимо искать пути ускорения оптимизации.

1. Предлагаемый метод понижения размерности

Предлагаемый подход заключается в предварительной кластеризации — разделении n доступных ценных бумаг на k групп (кластеров) ($k < n$). Кластеризация проводится иерархическим методом, в качестве матрицы расстояний используется матрица парных корреляций доходностей ценных бумаг.

Затем для каждого кластера рассчитывается доходность, как средняя доходностей входящих в него ценных бумаг, и строится ковариационная матрица доходностей кластеров. После этого можно будет решать задачу оптимизации портфеля с меньшим числом параметров, получив вектор долей для кластеров $W = [W_1, \dots, W_k]$. Рассчитать долю каждой ценной бумаги можно по формуле:

$$X_i = \frac{W_j}{S_j},$$

где

j — кластер, в который входит ценная бумага i;

S_j — число ценных бумаг в кластере j.

Из-за того, что кластеры представляют собой объединения ценных бумаг, оптимальный портфель, рассчитанный для кластеров, будет по своим характеристикам хуже, чем оптимальный портфель, рассчитанный для отдельных ценных бумаг. Величина снижения будет тем меньше, чем более схожие по поведению (с высокими коэффициентами парной корреляции) ценные бумаги оказались объединены в кластеры. Следовательно, можно сделать вывод о том, что для успешного применения предложенного метода (т.е. приводящего к достаточно сильному снижению размерности при допустимом снижении качества полученного оптимального решения) требуется, чтобы кластеры в исходных данных выделялись достаточно хорошо.

Возможно, результаты проведённой однажды кластеризации можно будет использовать в течение некоторого времени для решения нескольких задач оптимизации (до тех пор, пока значительно

не изменится матрица парных корреляций). Однако, этот вопрос однозначно требует дополнительного изучения.

Схожий с предложенным метод предлагается в работе [4], однако в упомянутом исследовании не производится расчёта ковариационной матрицы доходностей кластеров, а для оптимизации портфеля по модели Марковица в качестве матрицы σ используется полученная в ходе иерархической кластеризации матрица межкластерных корреляций. Такой подход, с одной стороны, ускоряет проведение расчётов, но с другой — авторами [4] признаётся риск того, что полученная матрица окажется отрицательно определённой (что сделает невозможным дальнейшие вычисления), однако данный риск игнорируется, поскольку он ни разу не реализовался в ходе экспериментов.

Подобная идея высказывается и в работе [5], выполненной в рамках проекта по исследованию оптимизации портфелей, основанной на кластеризации. Однако из-за слишком сильной ориентированности на практику (например, использование для оценки качества полученного оптимального решения метода Шарпа, специфичного для инвестиционных портфелей), результаты [5] сложно использовать для решения других задач оптимизации, кроме оптимизации инвестиционных портфелей.

2. Исследование влияния предлагаемого метода на качество получаемого оптимального решения

Для исследования влияния предлагаемого метода на качество получаемого оптимального решения был использован следующий метод:

Выбирались значения t_i — порогового значения для проведения кластеризации и R_{ic} — ожидаемой доходности. Затем с использованием предложенного метода при $t = t_i$ строился кластеризованный портфель с доходностью $R = R_{ic}$, его риск — V_i. Далее для тех же исходных данных, но без использования предложенного метода, строился портфель, оптимизированный по риску $V = V_i$, его доходность — $R_i u$.

Оптимизация портфелей проводилась по методике, описанной в [3], с использованием *Python* и библиотеки *CVXPY*, для кластеризации применялась библиотека *scipy.cluster*.

Для оценки влияния необходимы два показателя:

- $E = \frac{n}{k}$ — уровень снижения размерности в задаче оптимизации («экономия» размерности задачи)
- $L = R_i u - R_i c$ — снижение оптимизируемого показателя («потеря» оптимизируемого показателя)

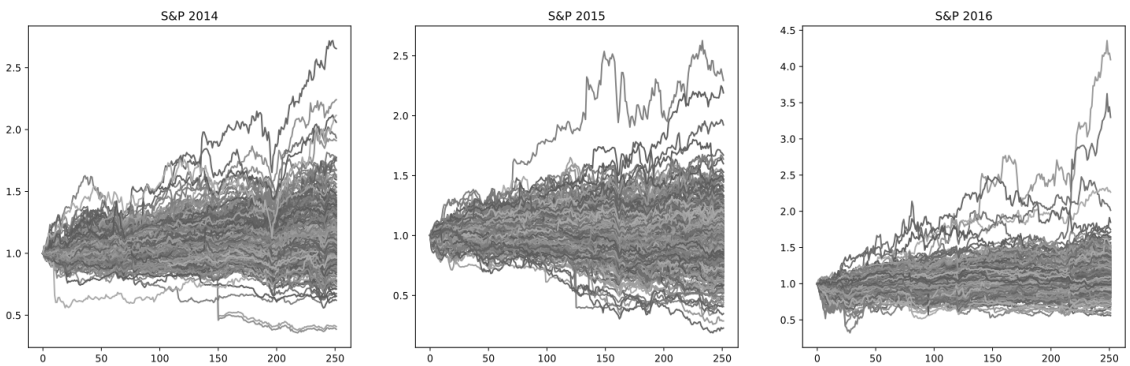


Fig. 1. Changes in prices of stocks of companies from the S&P rating in 2014-2016

Рис. 1. Изменение цен на акции компаний из рейтинга S&P в 2014-2016 годах

Эксперименты проводились на следующих данных:

1. Акции компаний из рейтинга Standard & Poor's 500 за 2014-2016 г.г. (данные об акциях 480, 486 и 494 компаний соответственно). Данные об изменениях цен на акции приведены на рисунке 1 (все данные нормированы).
2. Акции компаний, торгуемые на Нью-Йоркской фондовой бирже (NYSE) в 2004-2006 г.г. (данные об акциях 1285, 1354 и 1421 компаний соответственно). Данные об изменениях цен на акции приведены на рисунке 2 (все данные нормированы).

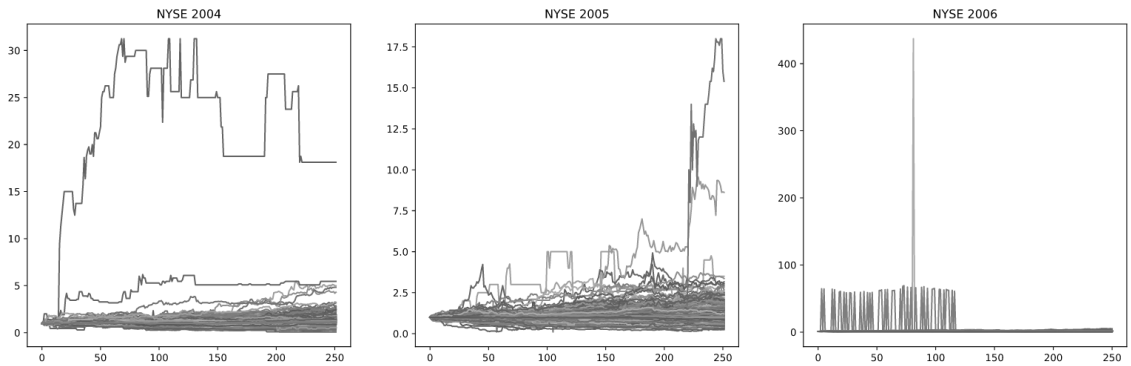


Fig. 2. Changes in prices of stocks of companies traded on the NYSE in 2004-2006

Рис. 2. Изменение цен на акции компаний, торгуемых на Нью-Йоркской фондовой бирже в 2004-2006 годах

3. Акции компаний, торгуемых на бирже NASDAQ в 2004-2006 г.г. (данные об акциях 1143, 1207 и 1280 компаний соответственно). Данные об изменениях цен на акции приведены на рисунке 3 (все данные нормированы).

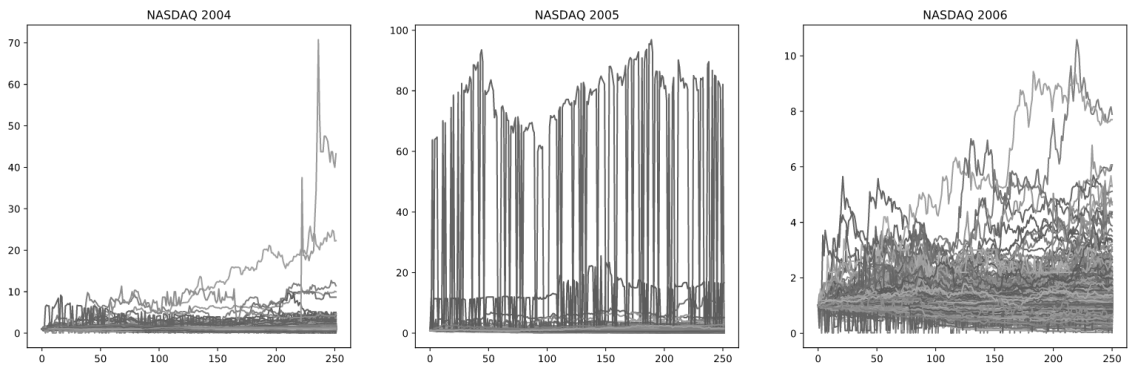


Fig. 3. Changes in prices of stocks of Companies traded on the NASDAQ in 2004-2006

Рис. 3. Изменение цен на акции компаний, торгуемых на бирже NASDAQ в 2004-2006 годах

Для проведения экспериментов использовались три основных метода иерархической агломеративной кластеризации: метод одиночной связи, метод полной связи и метод средней связи.

Число кластеров в зависимости от t для всех трёх наборов данных приведено на рисунках 4, 5, 6.

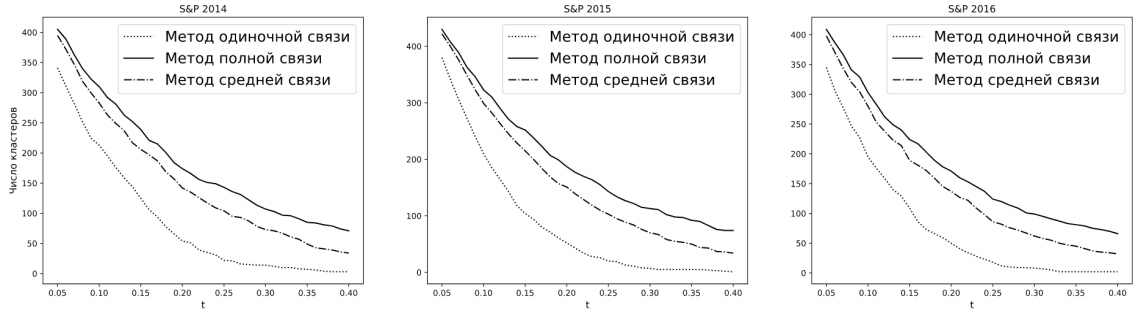


Fig. 4. Dependency between number of clusters and clustering threshold t for stocks of companies from the S&P rating

Рис. 4. Число кластеров в зависимости от порога кластеризации t для акций компаний из рейтинга S&P 500

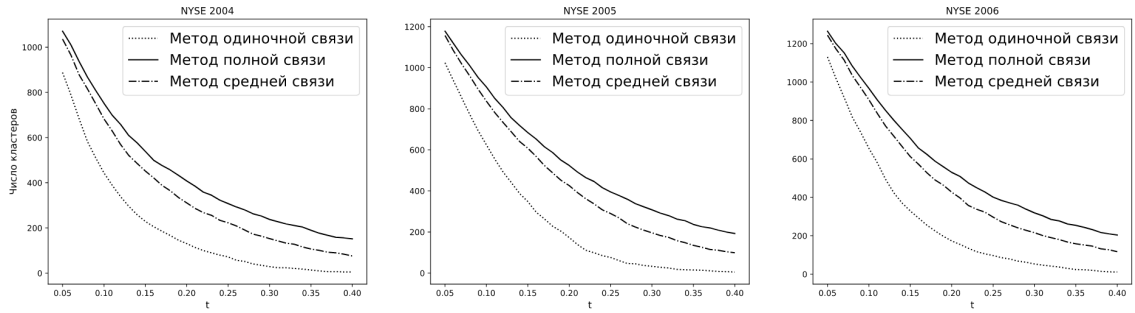


Fig. 5. Dependency between number of clusters and clustering threshold t for stocks of companies traded on the NYSE

Рис. 5. Число кластеров в зависимости от порога кластеризации t для акций компаний, торгуемых на Нью-Йоркской фондовой бирже

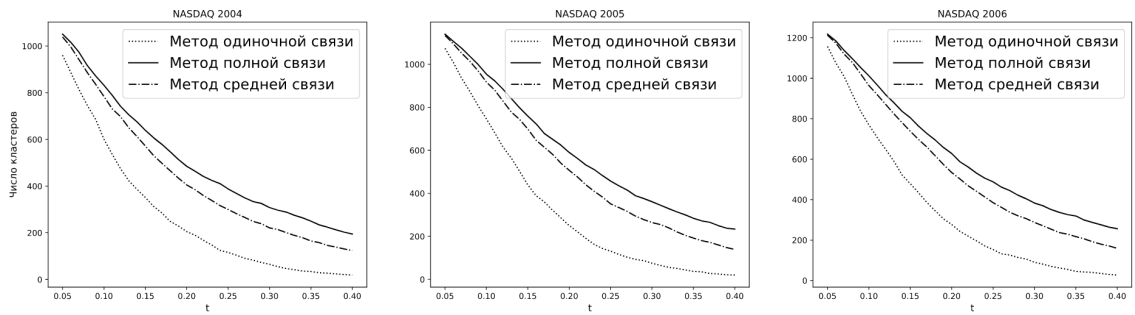


Fig. 6. Dependency between number of clusters and clustering threshold t for stocks of companies traded on the NASDAQ

Рис. 6. Число кластеров в зависимости от порога кластеризации t для акций компаний, торгуемых на бирже NASDAQ

2.1. Влияние предлагаемого метода на качество получаемого оптимального решения при кластеризации по методу одиночной связи

При кластеризации по методу одиночного соседа кластеры во всех трех наборах данных выделяются примерно одинаково, быстро и «гладко». В акциях компаний из рейтинга S&P кластеры выделяются при меньших пороговых значениях, чем в акциях компаний Нью-Йоркской фондовой биржи и акциях компаний биржи NASDAQ, что обусловлено, во-первых, меньшим объёмом данных, во-вторых, тем, что в рейтинг S&P попадают, в первую очередь, крупные компании, цены на акции которых достаточно стабильны и не показывают как значительного, «взрывного» роста, так и сильного снижения.

Средние результаты применения предлагаемого метода при кластеризации по методу одиночной связи приведены в таблице 1 (прочерк означает, что построить кластеризованный оптимальный портфель с заданным параметром доходности не удалось).

Table 1. Average results of the proposed method for clustering using the single linkage method

Таблица 1. Средние результаты применения предлагаемого метода при кластеризации по методу одиночной связи

t_i	R_{ic}	S&P 500 2014-2016		NYSE 2004-2006		NASDAQ 2004-2006	
		E	L	E	L	E	L
1	1,05	1,228	0,002	1,030	0,002	1,009	0,002
	1,1		0,007		0,000		0,000
	1,2		0,009		0,000		0,000
	1,3		0,009		0,000		0,000
2	1,05	2,990	0,018	1,575	0,002	1,403	0,007
	1,1		0,023		0,003		0,005
	1,2		0,036		0,007		0,007
	1,3		0,099		0,016		0,009
3	1,05	15,252	0,087	3,285	0,010	2,711	0,009
	1,1		0,105		0,021		0,015
	1,2		–		0,041		0,021
	1,3		–		0,060		0,027
4	1,05	87,694	0,202	8,399	0,022	7,564	0,038
	1,1		–		0,029		0,049
	1,2		–		0,051		0,074
	1,3		–		0,074		0,101

Как можно видеть, кроме очевидной зависимости E от t, существует ещё несколько зависимостей:

- L возрастает с ростом E при постоянном R_{ic}.
- L, в целом, возрастает с ростом R_{ic} при постоянном E.

Кроме того, можно отметить, что при меньшем объёме данных (акции компаний из рейтинга S&P 500) E растёт с ростом t быстрее, чем при большем объёме данных, и при некоторых значениях t и R_{ic} оптимальный портфель с заданными параметрами после кластеризации построить не удалось.

При значении $t = 1$ применение метода является практически бессмысленным, т.к. оно не приводит к существенному снижению размерности задачи оптимизации, а при $t = 2$ достаточно сильное снижение размерности происходит только для одного набора данных из трёх. В то же время, выбор $t = 4$ приводит к тому, что становится невозможно построить оптимальный портфель.

2.2. Влияние предлагаемого метода на качество получаемого оптимального решения при кластеризации по методу полной связи

Как можно видеть, объединение в кластеры при использовании метода полной связи происходит достаточно неравномерно и медленнее, чем при кластеризации по методу одиночной связи. В то же время, на всех трёх наборах данных кластеры выделяются достаточно хорошо, а для данных об акциях компаний Нью-Йоркской фондовой биржи и биржи NASDAQ — ещё и достаточно стабильно.

Средние результаты применения предлагаемого метода при кластеризации по методу полной связи приведены в таблице 2.

Table 2. Average results of the proposed method for clustering using the complete linkage method

Таблица 2. Средние результаты применения предлагаемого метода при кластеризации по методу полной связи

t_i	R_{ic}	S&P 500 2014-2016		NYSE 2004-2006		NASDAQ 2004-2006	
		E	L	E	L	E	L
1	1,05	1,125	0,002	1,025	0,001	1,009	0,002
	1,1		0,007		0,000		0,001
	1,2		0,009		0,000		0,000
	1,3		0,011		0,000		0,000
2	1,05	1,836	0,010	1,287	0,002	1,203	0,010
	1,1		0,014		0,003		0,010
	1,2		0,034		0,007		0,012
	1,3		0,050		0,014		0,020
3	1,05	3,175	0,026	1,836	0,005	1,713	0,023
	1,1		0,030		0,010		0,026
	1,2		0,043		0,019		0,033
	1,3		0,065		0,033		0,038
4	1,05	5,251	0,042	2,736	0,014	2,500	0,041
	1,1		0,046		0,021		0,046
	1,2		0,064		0,046		0,049
	1,3		0,103		0,099		0,058

Все зависимости, описанные для метода одиночной связи, имеют место и для метода полной связи, с единственным важным отличием — не было таких R_{ic} и t, при которых не получилось бы построить оптимальный портфель. Поскольку объединение в кластеры при использовании метода полной связи происходит медленее, чем при кластеризации по методу одиночной связи, в целом рост E и L при возрастании t и R_{ic} происходит медленнее, чем при использовании метода одиночной связи. Как и в случае с использованием метода одиночной связи, использование метода при $t = 1$ и $t = 2$ не имеет практического смысла, в то же время, наилучшие результаты получаются при выборе $t = 3$ или $t = 4$.

2.3. Влияние предлагаемого метода на качество получаемого оптимального решения при кластеризации по методу средней связи

При кластеризации по методу средней связи объединение в кластеры происходит немного более быстро и «гладко», чем при использовании метода полной связи, но всё же не так быстро, как при использовании метода одиночной связи. Относительно стабильные кластеры формируются только для акций компаний, торгуемых на бирже NASDAQ.

Средние результаты применения предлагаемого метода при кластеризации по методу средней связи приведены в таблице 3.

Table 3. Average results of the proposed method for clustering using the average linkage method

Таблица 3. Средние результаты применения предлагаемого метода при кластеризации по методу средней связи

t_i	R_{ic}	S&P 500 2014-2016		NYSE 2004-2006		NASDAQ 2004-2006	
		E	L	E	L	E	L
1	1,05	1,142	0,002	1,026	0,001	1,009	0,002
	1,1		0,007		0,000		0,000
	1,2		0,009		0,000		0,000
	1,3		0,011		0,000		0,000
2	1,05	2,054	0,011	1,343	0,002	1,233	0,014
	1,1		0,016		0,003		0,012
	1,2		0,030		0,008		0,010
	1,3		0,048		0,016		0,009
3	1,05	4,364	0,032	2,088	0,006	1,884	0,015
	1,1		0,037		0,010		0,019
	1,2		0,057		0,019		0,028
	1,3		0,096		0,033		0,040
4	1,05	9,354	0,065	3,511	0,015	3,173	0,025
	1,1		0,073		0,022		0,031
	1,2		0,099		0,049		0,044
	1,3		0,178		0,100		0,060

При кластеризации по методу средней связи результаты, во многом, обусловлены скоростью объединения акций в кластеры — средней между методами одиночной и полной связей. Как и в предыдущих случаях, использование метода при $t = 1$ лишено практического смысла, а наилучшие результаты получаются, в зависимости от размерности исходных данных, при выборе $t = 3$ или $t = 4$.

Заключение

По результатам описанных экспериментов можно сделать следующие выводы:

1. Предложенный метод позволяет существенно (в 5 раз и более) понижать размерность в задачах оптимизации при умеренном снижении качества полученного оптимального решения (до 5%) при «хорошем» выборе t и метода кластеризации.
2. Метод одиночной связи часто приводит к слишком быстрому понижению размерности, метод полной связи — к слишком медленному, «осторожному»; при использовании метода средней связи скорость будет средней. Выбор конкретного метода зависит, в первую очередь, от особенностей набора данных.

В дальнейшем следует изучить вопрос возможности построения регрессионной зависимости L от R, t и используемого метода кластеризации, чтобы формализовать процедуру выбора «хороших» параметров метода. Также потенциально перспективным является использование для проведения кластеризации ковариационной матрицы вместо матрицы парных корреляций, однако, этот вопрос требует отдельного изучения.

References

[1] H. Markowitz, "Portfolio Selection", *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[2] V. Dubrovin and O. Os'kiv, "Modeli i metody optimizacii vybora investicionnogo portfelja", *Radiojelektronika, informatika, upravlenie*, vol. 1, pp. 49–60, 2008.

[3] J. Chaitanya, *Markowitz Portfolio Optimization*, 2017. [Online]. Available: https://chaitjo.github.io/markowitz/.

[4] V. Tola, F. Lillo, M. Gallegati, and R. N. Mantegna, "Cluster analysis for portfolio optimization", *Journal of Economic Dynamics and Control*, vol. 32, no. 1, pp. 235–258, 2008.

[5] D. León and et. al., "Clustering algorithms for Risk-Adjusted Portfolio Construction", in *ICCS*, vol. 108, 2017, pp. 1334–1343.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

ALGORITHMS

Branch and Bound Algorithm for the Traveling Salesman Problem is not a Direct Type Algorithm

A. N. Maksimenko[1]

DOI: 10.18255/1818-1015-2020-1-72-85

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 90C57
Research article
Full text in Russian

Received December 3, 2019
After revision January 5, 2020
Accepted February 28, 2020

In this paper, we consider the notion of a direct type algorithm introduced by V. A. Bondarenko in 1983. A direct type algorithm is a linear decision tree with some special properties. The concept of a direct type algorithm is determined using the graph of solutions of a combinatorial optimization problem. The vertices of this graph are all feasible solutions of a problem. Two solutions are called adjacent if there are input data for which these and only these solutions are optimal. A key feature of direct type algorithms is that their complexity is bounded from below by the clique number of the solutions graph. In 2015-2018, there were five papers published, the main results of which are estimates of the clique numbers of polyhedron graphs associated with various combinatorial optimization problems. The main motivation in these works is the thesis that the class of direct type algorithms is wide and includes many classical combinatorial algorithms, including the branch and bound algorithm for the traveling salesman problem, proposed by J. D. C. Little, K. G. Murty, D. W. Sweeney, C. Karel in 1963. We show that this algorithm is not a direct type algorithm. Earlier, in 2014, the author of this paper showed that the Hungarian algorithm for the assignment problem is not a direct type algorithm. Thus, the class of direct type algorithms is not so wide as previously assumed.

Keywords: branch and bound; traveling salesman problem; linear decision tree; clique number; direct type algorithm

INFORMATION ABOUT THE AUTHORS

Aleksandr N. Maksimenko | orcid.org/0000-0002-0887-1500. E-mail: maximenko.a.n@gmail.com
PhD.

For citation: A. N. Maksimenko, "Branch and Bound Algorithm for the Traveling Salesman Problem is not a Direct Type Algorithm", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 72-85, 2020.

© Maksimenko A. N., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

ALGORITHMS

Алгоритм ветвей и границ для задачи коммивояжера не является алгоритмом прямого типа

А. Н. Максименко[1]

DOI: 10.18255/1818-1015-2020-1-72-85

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003, Россия.

УДК 519.16
Научная статья
Полный текст на русском языке

Получена 3 декабря 2019 г.
После доработки 5 января 2020 г.
Принята к публикации 28 февраля 2020 г.

В настоящей работе рассматривается понятие линейного разделяющего алгоритма прямого типа, введенное В. А. Бондаренко в 1983 г. Понятие алгоритма прямого типа определяется с помощью графа решений задачи комбинаторной оптимизации. Вершинами этого графа служат все допустимые решения задачи. Два решения называются смежными, если существуют входные данные, для которых эти решения и только они являются оптимальными. Ключевой особенностью алгоритмов прямого типа является то, что их трудоемкость оценивается снизу кликовым числом графа решений. В 2015–2018 гг. было опубликовано пять работ, основными результатами которых являются оценки кликовых чисел графов многогранников, ассоциированных с различными задачами комбинаторной оптимизации. В качестве основной мотивации в этих работах приводится тезис о том, что класс алгоритмов прямого типа является широким и включает в себя многие классические комбинаторные алгоритмы, в том числе алгоритм ветвей и границ для задачи коммивояжера, предложенный J. D. C. Little, K. G. Murty, D. W. Sweeney, C. Karel в 1963 г. Мы покажем, что этот алгоритм не является алгоритмом прямого типа. Ранее, в 2014 г., автором настоящей работы было показано, что венгерский алгоритм для задачи о назначениях не является алгоритмом прямого типа. Таким образом, класс алгоритмов прямого типа не является настолько широким, как предполагалось ранее.

Ключевые слова: метод ветвей и границ; задача коммивояжера; линейное разделяющее дерево; кликовое число; алгоритм прямого типа

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Николаевич Максименко | orcid.org/0000-0002-0887-1500. E-mail: maximenko.a.n@gmail.com
канд. физ.-мат. наук, доцент.

Для цитирования: A. N. Maksimenko, “Branch and Bound Algorithm for the Traveling Salesman Problem is not a Direct Type Algorithm”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 72-85, 2020.

© Максименко А. Н., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

В 2015–2018 гг. было опубликовано несколько работ [1—5], основными результатами которых являются оценки кликовых чисел графов многогранников, ассоциированных с различными задачами комбинаторной оптимизации. Основной мотивацией для таких оценок является следующий тезис: “It is known that this value characterizes the time complexity in a broad class of algorithms based on linear comparisons”[1] [5]. А именно, речь идет о классе алгоритмов прямого типа, впервые введенном в [6]. В качестве подтверждения этого тезиса в [2, 3] говорится о том, что этот класс включает алгоритмы сортировки, жадный алгоритм, динамическое программирование и метод ветвей и границ[2]. Доказательства того, что эти алгоритмы (а также алгоритм Эдмондса для задачи о паросочетаниях) являются алгоритмами прямого типа, впервые были опубликованы в диссертации [7] (см. также монографию [8]). В 2014 г. в [9] было показано, что алгоритм Куна—Манкреса для задачи о назначениях (а вместе с ним и алгоритм Эдмондса) не принадлежит к этому классу. Там же был описан часто используемый на практике способ модификации алгоритмов, выводящий их из класса алгоритмов прямого типа. Ниже мы докажем, что классический алгоритм ветвей и границ для задачи коммивояжера [10, 11] тоже не принадлежит к этому классу. Тем самым будет показано, что теорема 2.6.3 из диссертации [7] (теорема 3.6.6 из монографии [8]) не может быть доказана в оригинальной постановке. Это позволяет сделать вывод о том, что класс алгоритмов прямого типа не является столь широким, как предполагалось ранее.

Текст статьи организован следующим образом. В разделе 1 приводится псевдокод классического алгоритма ветвей и границ для задачи коммивояжера. В разделе 2 вводятся основные понятия концепции алгоритмов прямого типа и два ключевых определения: алгоритма прямого типа и алгоритма «прямого типа». В разделе 3 показано, что классический алгоритм ветвей и границ для задачи коммивояжера не является алгоритмом прямого типа, а в разделе 4 — что он не является алгоритмом «прямого типа».

1. Алгоритм ветвей и границ для задачи коммивояжера

Рассмотрим полный орграф $G = (V, A)$ с множеством вершин $V = [n] = \{1, 2, \dots, n\}$ и дуг $A = \{(i, j) \mid i, j \in V,\ i \neq j\}$. Каждой дуге $(i, j) \in A$ поставлено в соответствие число $c_{ij} \in \mathbb{Z}$, называемое *длиной дуги*. *Длиной подмножества* $H \subseteq A$ будем называть суммарную длину входящих в него дуг: $\mathrm{len}(H) = \sum_{(i,j)\in H} c_{ij}$. Задача коммивояжера состоит в том, чтобы найти $H^* \subseteq A$, являющееся гамильтоновым контуром в G и имеющее минимальную длину $\mathrm{len}(H^*)$.

Для удобства дальнейшего обсуждения поместим числа c_{ij} в матрицу $C = (c_{ij})$. Диагональным элементам c_{ii} припишем максимально возможные длины, $c_{ii} := \infty$, чтобы исключить их влияние на работу алгоритма, и будем предполагать, что $\infty - b = \infty$ для любого числа $b \in \mathbb{Z}$. Через $\mathrm{I}(M)$ будем обозначать множество индексов строк матрицы M, а через $\mathrm{J}(M)$ обозначим множество индексов столбцов матрицы M. В начале работы алгоритма $\mathrm{I}(C) = \mathrm{J}(C) = V$. Через $M(S, T)$ обозначим подматрицу матрицы M, лежащую на пересечении строк $S \subseteq \mathrm{I}(M)$ и столбцов $T \subseteq \mathrm{J}(M)$.

Сам алгоритм подробно описан в [11, раздел 4.1.6] и [10]. Мы приводим лишь его псевдокод — алгоритм 1. Отдельно, в алгоритме 2 описан процесс редуцирования строк и столбцов матрицы, а в алгоритме 3 — способ выбора такого нулевого элемента матрицы, при замене которого на бесконечность сумма редукций матрицы максимальна.

[1] «Известно, что эта величина характеризует сложность по времени в широком классе алгоритмов, основанных на линейных сравнениях»

[2] Но ссылки на источник с соответствующими доказательствами не приводятся.

Алгоритм 1. Метод ветвей и границ для задачи коммивояжера

Глобальные: гамильтонов контур Hopt с минимальной длиной; его длина lopt. До начала работы алгоритма lopt := ∞.

Вход : матрица длин M; множество дуг Arcs, обязательных для включения в контур; текущая сумма всех редукций sum. В самом начале работы алгоритма M := C, Arcs := $\emptyset$, sum := 0.

```
1  Procedure BranchBound(M, Arcs, sum)
       /* Редуцируем матрицу M                                          */
2      Reduction(M, sum)
3      if sum ≥ lopt then
4          завершить текущий экземпляр процедуры
       /* Выбираем оптимальный нулевой элемент матрицы M                */
5      (i, j) := ChooseArc(M)
       /* Разбираем случаи, когда контур содержит дугу (i, j)           */
6      if |I| = 3 then
           /* Находим единственный гамильтонов контур                   */
7          H := HamiltonCycle(Arcs ∪ {(i, j)})
8          if len(H) < lopt then
9              Hopt := H
10             lopt := len(H)
11     else
           /* Вычеркиваем i-ю строку и j-й столбец                      */
12         Mnew := M(I(M) \ {i}, J(M) \ {j})
           /* Находим запрещенную дугу                                  */
13         (l, k) := ForbiddenArc(Arcs,(i,j))
14         Mnew[l,k] := ∞
15         BranchBound(Mnew, Arcs ∪ {(i, j)}, sum)
       /* Разбираем случаи, когда контур не содержит дугу (i, j)        */
16     M[i,j] := ∞
17     BranchBound(M, Arcs, sum)

18 Function HamiltonCycle(Arcs)
19     Найти гамильтонов контур, содержащий все дуги из Arcs.

20 Function ForbiddenArc(Arcs,(i, j))
21     Найти пару вершин l и k, являющихся концом и началом наибольшего (по включению)
       пути в Arcs, содержащего (i, j).
```

Алгоритм 2. Редуцирование строк и столбцов матрицы

Вход : матрица M; текущая сумма всех редукций sum.
Выход : редуцированная матрица M; измененная sum.

```
1  Procedure Reduction(M, sum)
       /* Редуцируем строки матрицы M                                  */
2      for i ∈ I(M) do
3          m := ∞
           /* Находим  m = m(i) = min_{j∈J(M)} M[i,j]                  */
4          for j ∈ J(M) do
5              if m > M[i,j] then m := M[i,j]
6          sum := sum + m
7          for j ∈ J(M) do M[i,j] := M[i,j] − m
       /* Редуцируем столбцы матрицы M                                 */
8      for j ∈ J(M) do
9          m := ∞
10         for i ∈ I(M) do
11             if m > M[i,j] then m := M[i,j]
12         sum := sum + m
13         for i ∈ I(M) do M[i,j] := M[i,j] − m
```

Алгоритм 3. Выбор дуги

Вход : матрица M.
Выход : дуга (i^*, j^*), при запрещении которой нижняя оценка длины гамильтонова контура максимальна.

```
1  Function ChooseArc(M)
2      w := −1
3      for i ∈ I(M) do
4          for j ∈ J(M) do
5              if M[i,j] = 0 then
6                  m := ∞
                   /* Находим  m = min_t M[i,t]                       */
7                  for t ∈ J(M) \ {j} do
8                      if m > M[i,t] then m := M[i,t]
9                  k := ∞
                   /* Находим  k = min_t M[t,j]                       */
10                 for t ∈ I(M) \ {i} do
11                     if k > M[t,j] then k := M[t,j]
                   /* Сравниваем  m + k  с текущим рекордом  w        */
12                 if m + k > w then
13                     w := m + k
14                     (i*, j*) := (i, j)
```

2. Алгоритмы прямого типа

При изложении основ теории алгоритмов прямого типа мы будем придерживаться [7] (см. также [8]).

С целью унификации изложения матрица длин дуг C далее будет называться *вектором*[3] *входных данных* или просто *входом*. Решение задачи коммивояжера, т.е. гамильтонов контур $H \subseteq A$, будет представляться в виде 0/1-вектора $\boldsymbol{x} = (x_{ij})$, имеющего ту же размерность, что и C. Координаты этого вектора $x_{ij} = 1$, при $(i, j) \in H$, и $x_{ij} = 0$ иначе. Через X обозначаем множество всех 0/1-векторов $\boldsymbol{x}$, соответствующих гамильтоновым контурам в рассматриваемом орграфе G. Таким образом, при фиксированном входе C задача коммивояжера состоит в поиске решения $\boldsymbol{x}^* \in X$ такого, что $\langle \boldsymbol{x}^*, C\rangle \leq \langle \boldsymbol{x}, C\rangle \; \forall \boldsymbol{x} \in X$. Далее будем называть такое решение $\boldsymbol{x}^*$ *оптимальным относительно входа* C. Следуя [7, определение 1.1.2], совокупность всех таких оптимизационных задач, образованную фиксированным множеством допустимых решений X (в случае задачи коммивояжера, X однозначно определяется числом вершин орграфа G) и всевозможными входными векторами C, будем называть *задачей* X. Два допустимых решения $\boldsymbol{x}, \boldsymbol{y} \in X$ задачи X называются *смежными*, если найдется вектор C такой, что они, и только они, являются оптимальными относительно C. Подмножество $Y \subseteq X$ называется *кликой*, если любая пара $\boldsymbol{x}, \boldsymbol{y} \in Y$ смежна.

Выпуклая оболочка $\mathrm{conv}(X)$ называется *многогранником задачи* X. Так как X в задаче коммивояжера является подмножеством вершин единичного куба, то X совпадает с множеством вершин многогранника $\mathrm{conv}(X)$. В этой терминологии два решения $\boldsymbol{x}, \boldsymbol{y} \in X$ смежны тогда и только тогда, когда смежны соответствующие вершины многогранника $\mathrm{conv}(X)$ [7]. Известно [12], что все вершины многогранника коммивояжера попарно смежны при $n < 6$, где n — число вершин орграфа G, в котором требуется найти оптимальный гамильтонов контур.

Алгоритмы прямого типа относятся к классу линейных разделяющих алгоритмов, которые удобно представлять в виде линейных разделяющих деревьев.

Определение 1 ([7, определение 1.3.1])**.** *Линейным разделяющим деревом задачи* $X \subset \mathbb{Z}^m$ *называется ориентированное дерево, обладающее следующими свойствами:*

а) в каждый узел, за исключением одного, называемого корнем, входит ровно одна дуга; дуг, входящих в корень, нет;

б) для каждого узла либо имеется две выходящих из него дуги, либо таких дуг нет вообще; в первом случае узел называется внутренним, во втором — внешним, или листом;

в) каждому внутреннему узлу соответствует некоторый вектор $B \in \mathbb{Z}^m$*;*

г) каждому листу соответствует некоторый элемент из X *(нескольким листьям может соответствовать один и тот же элемент множества* X*);*

д) каждой дуге d *соответствует число* $\operatorname{sgn} d$*, равное* 1 *либо* -1*; две дуги, выходящие из одного узла, имеют различные значения;*

е) для каждой цепи $W = B_1 d_1 B_2 d_2 \ldots B_k d_k \boldsymbol{x}$*, соединяющей корень и лист (в обозначении цепи перечислены соответствующие ее узлам векторы* B_i*; дуга* d_i *выходит из узла* B_i*,* $i \in [k]$*), и для любого входа* C *из неравенств* $\langle B_i, C\rangle \operatorname{sgn} d_i \geq 0$*,* $i \in [k]$*, следует, что решение* $\boldsymbol{x}$ *является оптимальным относительно* C*.*

Таким образом, в рамках теории линейных разделяющих алгоритмов внимание уделяется только тем операциям, где выполняется проверка условий вида $\langle B, C\rangle \geq 0$, где C — вектор входных данных. Так, например, в строке 5 алгоритма 2 на самом первом шаге цикла проверяется неравенство $\infty > C_{11}$; на втором шаге проверяется условие $C_{11} > C_{12}$, и т. д. А в функциях `HamiltonCycle` и `ForbiddenArc` алгоритма 1, с точки зрения линейных разделяющих алгоритмов, не происходит

[3] Элементы матрицы всегда можно выписать в строку или столбец.

ничего интересного, так как не выполняются никакие сравнения с элементами вектора входных данных.

Процесс работы линейного разделяющего алгоритма для фиксированного вектора входных данных C представляет собой некоторую цепь $B_1 d_1 B_2 d_2 \dots B_m d_m \boldsymbol{x}$, соединяющую корень B_1 и некоторый лист $\boldsymbol{x}$ соответствующего линейного разделяющего дерева. Листом в нашем случае является гамильтонов контур (точнее, его характеристический вектор), являющийся оптимальным относительно C.

Пусть B — некоторый внутренний узел в линейном разделяющем дереве рассматриваемого алгоритма, а X — множество всех допустимых решений (множество меток всех листьев). Обозначим через X_B, $X_B \subseteq X$, множество меток всех листьев этого дерева, которым предшествует узел B, а через X_B^+ и X_B^- обозначим подмножества множества X_B, соответствующие двум выходящим из B дугам. Очевидно, $X_B = X_B^+ \cup X_B^-$. Обозначим через $R_B^- = X_B^+ \setminus X_B^-$ множество меток, отбрасываемых при переходе по «отрицательной» дуге. По аналогии определим множество меток $R_B^+ = X_B^- \setminus X_B^+$, отбрасываемых при переходе по «положительной» дуге.

Определение 2 ([7, определение 1.4.2]). *Линейное разделяющее дерево называется деревом* прямого типа*, если для любого внутреннего узла B и для любой клики $Y \subseteq X$ выполняется неравенство*

$$\min\{|R_B^+ \cap Y|, |R_B^- \cap Y|\} \le 1. \tag{1}$$

Непосредственно из определения следует, что высота дерева прямого типа (то есть число сравнений, используемых алгоритмом в худшем случае) для задачи X не может быть меньше, чем $\omega(X) - 1$, где $\omega(X)$ — кликовое число множества X [7, теорема 1.4.3].

Если же мы хотим доказать, что некий алгоритм не является алгоритмом прямого типа, достаточно указать клику Y, состоящую из четырех решений, и узел B такие, что $|R_B^+ \cap Y| = |R_B^- \cap Y| = 2$.

Для каждого $\boldsymbol{x} \in X$ определим *конус исходных данных*

$$K(\boldsymbol{x}) = \{C \mid \langle \boldsymbol{x}, C\rangle \le \langle \boldsymbol{y}, C\rangle,\ \forall \boldsymbol{y} \in X\}.$$

Т. е. $K(\boldsymbol{x})$ состоит из всех векторов C таких, что $\boldsymbol{x}$ оптимален относительно C.

Определение 3 ([7, определение 1.4.4]). *Линейное разделяющее дерево называется деревом* «прямого типа»*, если каждая цепь $B_1 d_1 B_2 d_2 \dots B_k d_k \boldsymbol{x}$, соединяющая корень и лист, удовлетворяет условиям:*

(*) для любого $\boldsymbol{y} \in X$, смежного с $\boldsymbol{x}$, найдется такой номер $i \in [k]$, что условия $\langle B_i, C\rangle \operatorname{sgn} d_i > 0$ и $C \in K(\boldsymbol{y})$ несовместны;

(**) для любого $i \in [k]$ из несовместности условий

$$\langle B_i, C\rangle \operatorname{sgn} d_i > 0 \qquad \text{и} \qquad C \in K(\boldsymbol{y})$$

для $\boldsymbol{y}$, смежного с $\boldsymbol{x}$, и из телесности конуса

$$K(\boldsymbol{x}) \cap \{C \mid \langle B_i, C\rangle \operatorname{sgn} d_i \le 0\}$$

следует, что ветвь, начинающаяся в узле B_i с дугой $-d_i$, имеет хотя бы один лист, помеченный $\boldsymbol{x}$.

Деревья «прямого типа» с деревьями прямого типа объединяет тот факт, что их высота тоже ограничена снизу величиной $\omega(X) - 1$ [7, теорема 1.4.5].

Чтобы доказать, что алгоритм 1 не является алгоритмом «прямого типа», мы ограничимся проверкой условия (*) из этого определения. А именно, мы укажем вполне конкретный входной вектор C^*, который однозначно определит некоторую цепь $B_1 d_1 B_2 d_2 \dots B_k d_k \boldsymbol{x}$. Далее будет выбран $\boldsymbol{y} \in X$, смежный с $\boldsymbol{x}$, для которого условия $\langle B_i, C\rangle \operatorname{sgn} d_i > 0$ и $C \in K(\boldsymbol{y})$ совместны при любом $i \in [k]$. Обратим особое внимание на то, что нам нужно будет проверить совместность условий $\langle B_i, C\rangle \operatorname{sgn} d_i > 0$ и $C \in K(\boldsymbol{y})$ отдельно для каждого $i \in [k]$, вне зависимости от результатов других сравнений. То есть для каждого $i \in [k]$ достаточно указать C_i такой, что $\langle B_i, C_i\rangle \operatorname{sgn} d_i > 0$ и $C_i \in K(\boldsymbol{y})$.

3. Алгоритм 1 не является прямым

Рассмотрим задачу коммивояжера в полном орграфе на 5 вершинах. Множество допустимых решений X такой задачи состоит из двадцати четырех 0/1-векторов, соответствующих гамильтоновым контурам в этом орграфе. Все 24 решения попарно смежны [12].

Предположим, что элементы матрицы длин дуг $C \in \mathbb{Z}^{5\times 5}$ удовлетворяют следующим условиям:

$$\begin{aligned} &c_{12} \le c_{13}, \quad c_{12} \le c_{14}, \quad c_{12} \le c_{15}, \\ &c_{21} \le c_{23}, \quad c_{21} \le c_{24}, \quad c_{21} \le c_{25}, \\ &c_{31} > c_{32}, \quad c_{32} > c_{34}, \quad c_{34} > c_{35}. \end{aligned} \tag{2}$$

В самом начале работы рассматриваемого алгоритма выполняется процедура редуцирования этой матрицы (алгоритм 2). Мы ограничимся рассмотрением этапа редуцирования строк. В результате последовательных сравнений в первой строке выбирается наименьший элемент (в данном случае c_{12}) и вычитается из всех её элементов. Далее выбирается минимальный элемент во второй строке, им оказывается c_{21}, и минимальный элемент в третьей строке — c_{35}. После этого алгоритм переходит к проверке неравенства

$$c_{41} > c_{42} \tag{3}$$

(сравнение $\infty > c_{41}$ присутствует в алгоритме исключительно для краткости описания и не несет никакой информации). Соответствующий узел линейного разделяющего дерева алгоритма обозначим B. Ясно, что алгоритм попадает в этот узел дерева, если, и только если для входного вектора C выполняются условия (2).

Рассмотрим характеристические вектора четырех гамильтоновых контуров:

$$\boldsymbol{x} = \begin{pmatrix} & 0 & 0 & 1 & 0 \\ 0 & & 0 & 0 & 1 \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & 0 & \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} & 0 & 0 & 0 & 1 \\ 0 & & 1 & 0 & 0 \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 0 & 1 & \end{pmatrix},$$

$$\boldsymbol{z} = \begin{pmatrix} & 0 & 1 & 0 & 0 \\ 0 & & 0 & 0 & 1 \\ 0 & 1 & & 0 & 0 \\ 1 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 1 & \end{pmatrix}, \quad \boldsymbol{w} = \begin{pmatrix} & 0 & 0 & 0 & 1 \\ 0 & & 0 & 1 & 0 \\ 0 & 1 & & 0 & 0 \\ 1 & 0 & 0 & & 0 \\ 0 & 0 & 1 & 0 & \end{pmatrix}.$$

Нетрудно проверить, что входные векторы

$$C_x = \begin{pmatrix} & 0 & 6 & 1 & 6 \\ 0 & & 6 & 6 & 1 \\ 3 & 2 & & 1 & 0 \\ 6 & 0 & 6 & & 6 \\ 6 & 6 & 0 & 6 & \end{pmatrix}, \quad C_y = \begin{pmatrix} & 0 & 6 & 6 & 1 \\ 0 & & 1 & 6 & 6 \\ 3 & 2 & & 1 & 0 \\ 6 & 0 & 6 & & 6 \\ 6 & 6 & 6 & 0 & \end{pmatrix},$$

$$C_z = \begin{pmatrix} & 0 & 1 & 6 & 6 \\ 0 & & 6 & 6 & 1 \\ 6 & 3 & & 1 & 0 \\ 0 & 6 & 6 & & 6 \\ 6 & 6 & 6 & 0 & \end{pmatrix}, \quad C_w = \begin{pmatrix} & 0 & 6 & 6 & 1 \\ 0 & & 6 & 1 & 6 \\ 6 & 3 & & 1 & 0 \\ 0 & 6 & 6 & & 6 \\ 6 & 6 & 0 & 6 & \end{pmatrix}$$

удовлетворяют условиям (2), а для каждого $\boldsymbol{t} \in \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w}\}$ и для любого $\boldsymbol{s} \in X \setminus \{\boldsymbol{t}\}$ выполняется неравенство $\langle \boldsymbol{t}, C_t \rangle = 5 < \langle \boldsymbol{s}, C_t \rangle$. Следовательно, все четыре вектора входят в множество меток X_B всех листьев дерева алгоритма, которым предшествует узел B.

Покажем, что $\boldsymbol{z}$ и $\boldsymbol{w}$ входят в множество меток R_B^+, отбрасываемых при выполнении неравенства (3), а $\boldsymbol{x}$ и $\boldsymbol{y}$ входят в множество меток R_B^-, отбрасываемых при невыполнении неравенства (3).

Предположим, что для входной матрицы C выполнены условия (2) и неравенство (3). Тогда $\langle \boldsymbol{z}, C \rangle > \langle \boldsymbol{z}', C \rangle$ для

$$\boldsymbol{z}' = \begin{pmatrix} & 0 & 1 & 0 & 0 \\ 1 & & 0 & 0 & 0 \\ 0 & 0 & & 0 & 1 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 0 & 1 & \end{pmatrix}.$$

Аналогично, $\langle \boldsymbol{w}, C \rangle > \langle \boldsymbol{w}', C \rangle$ для

$$\boldsymbol{w}' = \begin{pmatrix} & 0 & 0 & 0 & 1 \\ 1 & & 0 & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & 0 & \end{pmatrix}.$$

Таким образом, $\boldsymbol{z}, \boldsymbol{w} \in R_B^+$.

Предположим, что для C выполнены условия (2), но не выполнено неравенство (3). Тогда $\langle \boldsymbol{x}, C \rangle > \langle \boldsymbol{x}', C \rangle$ для

$$\boldsymbol{x}' = \begin{pmatrix} & 1 & 0 & 0 & 0 \\ 0 & & 0 & 0 & 1 \\ 0 & 0 & & 1 & 0 \\ 1 & 0 & 0 & & 0 \\ 0 & 0 & 1 & 0 & \end{pmatrix},$$

и $\langle \boldsymbol{y}, C \rangle > \langle \boldsymbol{y}', C \rangle$ для

$$\boldsymbol{y}' = \begin{pmatrix} & 1 & 0 & 0 & 0 \\ 0 & & 1 & 0 & 0 \\ 0 & 0 & & 0 & 1 \\ 1 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 1 & \end{pmatrix}.$$

Следовательно, $\boldsymbol{z}, \boldsymbol{w} \in R_B^+$.

Таким образом, условие (1) для данного узла B не выполнено, и алгоритм 1 не является алгоритмом прямого типа.

4. Алгоритм 1 не является «прямым»

При анализе алгоритма 1, как линейного разделяющего дерева, нам будут встречаться только неравенства следующего вида:

$$\langle B^+, C \rangle - \langle B^-, C \rangle > 0, \tag{4}$$

где $C \in \mathbb{Z}^{n^2}$ — вектор входных данных,

$$B^+, B^- \in \{0,1\}^{n^2}, \quad \langle B^+, B^- \rangle = 0 \quad \text{и} \quad \langle B^+, \mathbf{1} \rangle = \langle B^-, \mathbf{1} \rangle > 0, \tag{5}$$

$\mathbf{1}$ — вектор из единиц. Иными словами, условие (5) означает, что множества единичных координат для B^+ и B^- равномощны и не пересекаются. Для каждого такого неравенства и для некоторого

допустимого решения $\boldsymbol{y} \in X \subset \{0,1\}^{n^2}$ нам нужно будет проверить, что существует $C \in K(\boldsymbol{y})$, для которого это неравенство выполнено. Такой анализ существенно упрощается, если воспользоваться следующим критерием.

Лемма 1. *Пусть $\boldsymbol{y} \in \{0,1\}^{n^2}$ — характеристический вектор некоторого гамильтонова контура в полном орграфе $G = ([n], A)$. Если выполняются условия (5) и $\langle B^+, \boldsymbol{y}\rangle \le 2$, то неравенство (4) и условие $C \in K(\boldsymbol{y})$ совместны.*

Доказательство. Пусть

$$S = \{(i,j) \in [n]^2 \mid y_{ij} = 1 \text{ и } B^+_{ij} = 0\}.$$

Из условия $\langle B^+, \boldsymbol{y}\rangle \le 2$ следует, что $|S| \ge n-2$. Положим

$$C := \mathbf{4} - B^-$$

и, после этого, $C_{ij} := 0$ для $(i,j) \in S$.
Тогда $\langle B^+, C\rangle = \langle B^+, \mathbf{4} - B^-\rangle = \langle B^+, \mathbf{4}\rangle$ и $\langle B^-, C\rangle \le \langle B^-, \mathbf{4} - B^-\rangle = \langle B^+, \mathbf{4}\rangle - \langle B^-, B^-\rangle$ (так как B^+ и B^- удовлетворяют условиям (5)). Следовательно, неравенство (4) для такого C будет выполнено.

Покажем теперь, что $\langle \boldsymbol{y}, C\rangle < \langle \boldsymbol{x}, C\rangle$ для любого $\boldsymbol{x} \in X \setminus \boldsymbol{y}$.

Очевидно, $\langle \boldsymbol{y}, C\rangle = (n - |S|)4 \le 8$.

Пусть $\boldsymbol{x} \in X$. Заметим, что если $\langle \boldsymbol{y}, \boldsymbol{x}\rangle \ge n-2$, то $\boldsymbol{x} = \boldsymbol{y}$, так как любой гамильтонов контур в орграфе на n вершинах однозначно определяется по любым своим $n-2$ дугам. Следовательно, $\langle \boldsymbol{x}, C\rangle \ge 3 \cdot 3 = 9$ для любого $\boldsymbol{x} \in X \setminus \boldsymbol{y}$. □

В частности, условия леммы выполнены, если в B^+ не более двух единиц.

Итак, положим $n = 4$ и рассмотрим следующий вектор входных данных (вместо бесконечности будем подставлять пробел):

$$C^* := \begin{pmatrix} & 0 & 2 & 1 \\ 2 & & 0 & 2 \\ 1 & 2 & & 0 \\ 0 & 1 & 2 & \end{pmatrix}. \tag{6}$$

Ясно, что единственным оптимальным решением будет вектор

$$\boldsymbol{x} := \begin{pmatrix} & 1 & 0 & 0 \\ 0 & & 1 & 0 \\ 0 & 0 & & 1 \\ 1 & 0 & 0 & \end{pmatrix}$$

и соответствующий ему контур $\{(1,2),(2,3),(3,4),(4,1)\}$. Нетрудно проверяется, что множество всех допустимых решений X состоит из 6 попарно смежных векторов. Положим

$$\boldsymbol{y} := \begin{pmatrix} & 0 & 0 & 1 \\ 0 & & 1 & 0 \\ 1 & 0 & & 0 \\ 0 & 1 & 0 & \end{pmatrix}.$$

Обратим внимание, что $\boldsymbol{y}$ является вторым (после $\boldsymbol{x}$) по оптимальности относительно C^*. Именно это обстоятельство во многом упрощает дальнейшую проверку соответствующих сравнений.

В целом схема работы алгоритма при заданном входе C^* изображена на рис. 1.

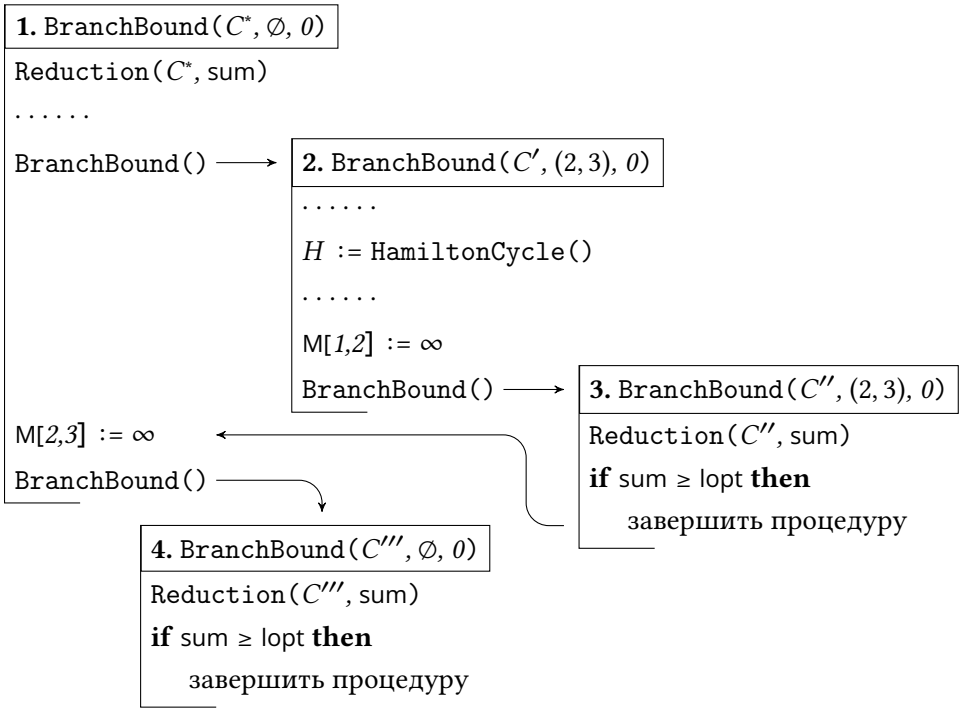


Fig. 1. General scheme of work of the algorithm 1 for the input given by the formula (6)

Рис. 1. Общая схема работы алгоритма 1 для входа, задаваемого формулой (6)

Рассмотрим, прежде всего, какие неравенства проверяются при первом входе в процедуру BranchBound с входом C^*. При редуцировании первой строки матрицы C^* (строка 5 алгоритма 2) проверяются (и выполняются) неравенства $\infty > C_{12}$, $C_{13} > C_{12}$ и $C_{14} > C_{12}$. Далее мы не будем рассматривать неравенства, в которых сумма (либо разность) элементов исходной матрицы сравнивается с бесконечностью, так как они всегда выполняются и совместны с любым допустимым решением. Заметим, что только что перечисленные неравенства удовлетворяют условиям леммы 1, так как $\langle B^+, \mathbf{1}\rangle = 1$. А значит, они совместны с условием $C \in K(\boldsymbol{y})$.

После редуцирования первой строки в её ячейках $\mathsf{M}[1, j]$, $j \in [4]$, содержатся разности $C_{1j} - C_{12}$, а переменная sum принимает значение C_{12}.

При редуцировании второй строки проверяются неравенства $C_{21} > C_{23}$ и $C_{24} > C_{23}$. Согласно лемме 1, они совместны с условием $C \in K(\boldsymbol{y})$.

После редуцирования второй строки в её ячейках $\mathsf{M}[2, j]$, $j \in [4]$, содержатся разности $C_{2j} - C_{23}$, а переменная sum принимает значение $C_{12} + C_{23}$.

При редуцировании последних двух строк ситуация полностью аналогична. После завершения редуцирования строк

$$\mathsf{sum} = C_{12} + C_{23} + C_{34} + C_{41},$$

$$\mathsf{M} = \begin{pmatrix} & 0 & C_{13} - C_{12} & C_{14} - C_{12} \\ C_{21} - C_{23} & & 0 & C_{24} - C_{23} \\ C_{31} - C_{34} & C_{32} - C_{34} & & 0 \\ 0 & C_{42} - C_{41} & C_{43} - C_{41} & \end{pmatrix}.$$

Далее, при редуцировании первого столбца проверяются неравенства $\mathsf{M}[2,1] > \mathsf{M}[3,1]$ и $\mathsf{M}[3,1] > \mathsf{M}[4,1]$. Нам известно, что $\mathsf{M}[2,1] = C_{21} - C_{23}$, $\mathsf{M}[3,1] = C_{31} - C_{34}$, $\mathsf{M}[4,1] = C_{41} - C_{41} = 0$. Следовательно, проверяются неравенства $C_{21} - C_{23} > C_{31} - C_{34}$ и $C_{31} - C_{34} > 0$. Каждое из них удовлетворяет условиям леммы 1.

При редуцировании оставшихся трех столбцов ситуация повторяется. Значение sum при редуцировании столбцов не меняется, так как каждый столбец уже содержит нули.

После этого в алгоритме 1 выполняется проверка условия sum ≥ lopt. Но lopt = ∞. Поэтому алгоритм переходит к вычислению функции `ChooseArc`.

Первым нулевым элементом является M[1, 2]. После этого в строке 8 алгоритма 3 выполняются сравнения ∞ > M[1, 3] и M[1, 3] > M[1, 4]. При этом, после предыдущего этапа редукции, имеем M[1, 3] = $C_{13} - C_{12}$ и M[1, 4] = $C_{14} - C_{12}$. Очевидно, неравенство $C_{13} - C_{12} > C_{14} - C_{12}$ удовлетворяет условиям леммы 1. На этом шаге выполняется присвоение $m := C_{14} - C_{12}$. Далее, в строке 11 алгоритма 3 выполняются сравнения ∞ > M[3, 2] и M[3, 2] > M[4, 2]. При этом M[3, 2] = $C_{32} - C_{34}$ и M[4, 2] = $C_{42} - C_{41}$. Условия леммы 1 снова выполнены. На этом шаге выполняется присвоение $k := C_{42} - C_{41}$. Далее выполняется сравнение $m + k > -1$ или, что то же самое, $C_{14} - C_{12} + C_{42} - C_{41} > -1$. Очевидно, это неравенство совместимо с условием $C \in K(\boldsymbol{y})$. В переменную w заносится значение выражения $C_{14} - C_{12} + C_{42} - C_{41}$.

Второй нулевой элемент — M[2, 3]. Действуя по аналогии, перечислим только нетривиальные сравнения. Неравенство M[2, 1] ≤ M[2, 4] или $C_{21} - C_{23} \le C_{24} - C_{23}$, очевидно, совместимо с условием $C \in K(\boldsymbol{y})$. Неравенство M[1, 3] ≤ M[4, 3] тоже совместимо. Далее, в строке 12 проверяется неравенство $m + k > w$ или, с учетом предыдущих действий,

$$C_{21} - C_{23} + C_{13} - C_{12} > C_{14} - C_{12} + C_{42} - C_{41}.$$

Очевидно, оно удовлетворяет условиям леммы 1. После этого шага

$$w = C_{21} - C_{23} + C_{13} - C_{12}.$$

Третий нулевой элемент — M[3, 4]. Неравенство M[3, 1] < M[3, 2] или $C_{31} - C_{34} < C_{32} - C_{34}$, очевидно, совместимо с условием $C \in K(\boldsymbol{y})$. Неравенство M[1, 4] < M[2, 4] тоже совместимо. Условие $m + k < w$ имеет вид

$$C_{14} - C_{12} + C_{31} - C_{34} < C_{21} - C_{23} + C_{13} - C_{12}$$

и тоже совместимо с условием $C \in K(\boldsymbol{y})$.

Четвертый нулевой элемент — M[4, 1]. Легко проверить, что M[4, 2] < M[4, 3] и M[3, 1] < M[2, 1] совместимы с условием $C \in K(\boldsymbol{y})$. Условие $m + k < w$ имеет вид

$$C_{31} - C_{34} + C_{42} - C_{41} < C_{21} - C_{23} + C_{13} - C_{12}$$

и тоже совместимо.

В данный момент мы все еще находимся в первом экземпляре процедуры `BranchBound`. После описанного выше выполнения функции `ChooseArc` выбирается дуга $(i, j) = (2, 3)$ (сумма $m + k$ для нее оказалась наибольшей), из матрицы M вычеркиваются 2-я строка и 3-й столбец, а дуга (3, 2) становится запрещенной. На вход второго экземпляра процедуры `BranchBound` подается матрица

$$C' := \begin{pmatrix} & 0 & 1 \\ & & \\ 1 & & 0 \\ 0 & 1 & \end{pmatrix}$$

(пустая строка и пустой столбец оставлены для удобства чтения). Ясно, что при её редуцировании ничего нового не происходит, так как каждая строка и каждый столбец содержат нули. При вызове функции `ChooseArc` в строке 12 выполняются следующие сравнения типа $m + k > w$.

$$C_{14} - C_{12} + C_{42} - C_{41} > -1.$$

Очевидно, это неравенство совместимо с условием $C \in K(\boldsymbol{y})$. Далее, выполняется неравенство

$$C_{31} - C_{34} + C_{14} - C_{12} \le C_{14} - C_{12} + C_{42} - C_{41},$$

которое удовлетворяет условиям леммы 1. Следующее сравнение

$$C_{31} - C_{34} + C_{42} - C_{41} \le C_{14} - C_{12} + C_{42} - C_{41}$$

тоже совместимо с $C \in K(\boldsymbol{y})$.

Итак, после вызова функции `ChooseArc` во втором экземпляре `BranchBound`, выбирается дуга $(1, 2)$. Гамильтонов цикл с дугами $(2, 3)$ и $(1, 2)$ определяется однозначно. Выполняется присвоение

$$\mathsf{lopt} := C_{12} + C_{23} + C_{34} + C_{41}.$$

После этого алгоритм переходит к рассмотрению случаев, когда контур содержит дугу $(2, 3)$, но не содержит $(1, 2)$. Запускается третий экземпляр `BranchBound` с матрицей

$$C'' := \begin{pmatrix} & & & 1 \\ & & & \\ 1 & & & 0 \\ 0 & 1 & & \end{pmatrix}.$$

При редуцировании две единицы заменяются нулями. Никакие «отбрасывающие» сравнения не выполняются. Значение переменной sum увеличивается на $\mathsf{M}[1, 4] = C_{14} - C_{12}$ и на $\mathsf{M}[4, 2] = C_{42} - C_{41}$. Текущий экземпляр процедуры завершается в строке 3 после проверки неравенства $\mathsf{sum} \ge \mathsf{lopt}$:

$$(C_{14} - C_{12}) + (C_{42} - C_{41}) > 0.$$

Заметим, что допустимое решение $\boldsymbol{y}$ полностью отбраковывается алгоритмом именно на этом шаге (с учетом ранее проверенного неравенства $C_{31} > C_{34}$). Тем не менее, это неравенство удовлетворяет условиям леммы 1 и, следовательно, совместно с условием $C \in K(\boldsymbol{y})$.

Вместе с третьим экземпляром процедуры `BranchBound` завершается и второй её экземпляр. Алгоритм переходит к выполнению предпоследней строки в первом экземпляре. В этом экземпляре

$$\mathsf{sum} = C_{12} + C_{23} + C_{34} + C_{41}.$$

Для разбора случаев, когда контур не содержит дугу $(2, 3)$, вызывается четвертый экземпляр процедуры с матрицей

$$C''' := \begin{pmatrix} & 0 & 2 & 1 \\ 2 & & & 2 \\ 1 & 2 & & 0 \\ 0 & 1 & 2 & \end{pmatrix}.$$

При редуцировании второй строки выполняется сравнение $\mathsf{M}[2, 1] \le \mathsf{M}[2, 4]$. При редуцировании третьего столбца — $\mathsf{M}[1, 3] \le \mathsf{M}[4, 3]$. Очевидно, ни то ни другое не отбрасывают целиком конус $K(\boldsymbol{y})$. Значение sum увеличивается на $(C_{21} - C_{23}) + (C_{13} - C_{12})$.

И, наконец, сравнение $\mathsf{sum} \ge \mathsf{lopt}$ завершает этот четвертый экземпляр процедуры и вообще весь алгоритм. Это сравнение имеет вид

$$(C_{21} - C_{23}) + (C_{13} - C_{12}) \ge 0$$

и тоже совместимо с условием $C \in K(\boldsymbol{y})$.

Итак, условие (*) из определения 3 не выполнено для этого алгоритма.

References

[1] V. Bondarenko, A. Nikolaev, and D. Shovgenov, "1-skeletons of the spanning tree problems with additional constraints", *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 682–688, 2017.

[2] V. Bondarenko and A. Nikolaev, "On graphs of the cone decompositions for the min-cut and max-cut problems", *International Journal of Mathematics and Mathematical Sciences*, vol. 2016, 2016.

[3] V. Bondarenko and A. Nikolaev, "Some properties of the skeleton of the pyramidal tours polytope", *Electronic Notes in Discrete Mathematics*, vol. 61, pp. 131–137, 2017.

[4] V. A. Bondarenko, A. V. Nikolaev, and D. Shovgenov, "Polyhedral characteristics of balanced and unbalanced bipartite subgraph problems", *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 576–585, 2017.

[5] V. Bondarenko and A. Nikolaev, "On the skeleton of the polytope of pyramidal tours", *Journal of Applied and Industrial Mathematics*, vol. 12, no. 1, pp. 9–18, 2018.

[6] V. Bondarenko, "Nonpolynomial lowerbound of the traveling salesman problem complexity in one class of algorithms", *Automation and Remote Control*, vol. 44, no. 9, pp. 1137–1142, 1983.

[7] V. Bondarenko, "Geometricheskie metody sistemnogo analiza v kombinatornoy optimizatsii", *diss. … dokt. fiz.-mat. nauk, Yaroslavl*, 1993.

[8] V. Bondarenko and A. Maksimenko, *Geometricheskie konstruktsii i slozhnost v kombinatornoy optimizatsii*. Moskva: URSS, 2008, 182 pp.

[9] A. Maksimenko, "Kharakteristiki slozhnosti: klikovoe chislo grafa mnogogrannika i chislo pryamougolnogo pokrytiya", *Modelirovanie i analiz informatsionnykh sistem*, vol. 21, no. 5, pp. 116–130, 2014.

[10] J. Little, K. Murty, D. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem", *Operations research*, vol. 11, no. 6, pp. 972–989, 1963.

[11] E. Reingold, J. Nievergelt, and N. Deo, *Combinatorial algorithms: theory and practice*. Pearson College Div, 1977, 433 pp.

[12] M. Padberg and M. Rao, "The travelling salesman problem and a class of polyhedra of diameter two", *Mathematical Programming*, vol. 7, no. 1, pp. 32–45, 1974.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

SOFTWARE

Parallel Algorithm for Solving the Graph Isomorphism Problem

V. V. Vasilchikov[1]

DOI: 10.18255/1818-1015-2020-1-86-94

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 68W10
Research article
Full text in Russian

Received January 16, 2020
After revision February 24, 2020
Accepted February 28, 2020

In this paper, we offer an efficient parallel algorithm for solving the Graph Isomorphism Problem. Our goal is to construct a suitable vertex substitution or to prove the absence of such. The problem is solved for undirected graphs without loops and multiple edges, it is assumed that the graphs can be disconnected. The question of the existence or absence of an algorithm for solving this problem with polynomial complexity is currently open. Therefore, as for any time-consuming task, the question arises of accelerating its solution by parallelizing the algorithm. We used the RPM_ParLib library developed by the author as the main tool to program the algorithm. This library allows us to develop effective applications for parallel computing on a local network in the .NET Framework. Such applications have the ability to generate parallel branches of computation directly during program execution and dynamically redistribute work between computing modules. Any language with support for the .NET Framework can be used as a programming language in conjunction with this library. For our experiments, we developed some C# applications using this library. The main purpose of these experiments was to study the acceleration achieved by recursive-parallel computing. Specially generated random regular graphs with varying degrees of vertices were used as initial data. A detailed description of the algorithm and its testing, as well as the results obtained, are also given in the paper.

Keywords: graph isomorphism problem; parallel algorithm; recursion; .NET

INFORMATION ABOUT THE AUTHORS

Vladimir Vasilyevich Vasilchikov | orcid.org/0000-0001-7882-8906. E-mail: vvv193@mail.ru
PhD.

Funding: This work was supported by initiative program VIP-004 (state registration number AAAA-A16-116070610022-6).

For citation: V. V. Vasilchikov, “Parallel Algorithm for Solving the Graph Isomorphism Problem”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 86-94, 2020.

© Vasilchikov V. V., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

SOFTWARE

Параллельный алгоритм решения задачи об изоморфизме графов

В. В. Васильчиков[1]

DOI: 10.18255/1818-1015-2020-1-86-94

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003 Россия.

УДК 519.688: 519.85
Научная статья
Полный текст на русском языке

Получена 16 января 2020 г.
После доработки 24 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

В данной работе предлагается параллельный алгоритм решения задачи об изоморфизме графов. Целевым результатом для нас выступает построение подходящей подстановки вершин, либо доказательство отсутствия таковой. Задача решается для неориентированных графов без петель и кратных ребер, допускается, что графы могут быть несвязными. Вопрос о существовании либо отсутствии алгоритма с полиномиальной трудоемкостью в настоящее время является открытым. Следовательно, как и для любой трудоемкой задачи, возникает вопрос об ускорении ее решения за счет распараллеливания алгоритма. Для организации параллельных вычислений автором использовалась библиотека RPM_ParLib, которая позволяет создавать параллельные приложения, работающие в локальной вычислительной сети под управлением среды исполнения .NET Framework. Библиотека поддерживает рекурсивно-параллельный стиль программирования и обеспечивает эффективное распределение работы и динамическую балансировку загрузки вычислительных модулей в процессе исполнения программы. Она может быть использована для приложений, написанных на любом языке программирования, поддерживаемом .NET Framework. Для решения нашей задачи и проведения численного эксперимента было разработано несколько приложений на языке C#. Целью эксперимента было исследование ускорения, достигаемого за счет рекурсивно-параллельной организации вычислений. В качестве исходных данных использовались специально сгенерированные случайные регулярные графы с различной степенью вершин. Подробное описание алгоритма и эксперимента, а также полученные результаты также приводятся в работе.

Ключевые слова: изоморфизм графов; параллельный алгоритм; рекурсия; .NET

ИНФОРМАЦИЯ ОБ АВТОРАХ

Владимир Васильевич Васильчиков | orcid.org/0000-0001-7882-8906. E-mail: vvv193@mail.ru
канд. техн. наук, зав. кафедрой вычислительных и программных систем.

Финансирование: Работа выполнена в рамках инициативной НИР ВИП-004
(номер госрегистрации АААА-А16-116070610022-6).

Для цитирования: V. V. Vasilchikov, “Parallel Algorithm for Solving the Graph Isomorphism Problem”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 86-94, 2020.

© Васильчиков В. В., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

Задача об изоморфизме графов является одной из классических задач дискретной оптимизации [1]. Для нее пока не доказана принадлежность ни к классу P, ни к классу NP-полных задач. Потребность в решении этой задачи возникает в самых разных предметных областях, где требуется установление идентичности структур тех или иных сложных систем. В качестве примеров можно назвать транспортные, энергетические системы, системы связи, электронные схемы, системы распознавания образов, а также задачи математической химии, исследование социальных сетей и многие другие.

Поскольку для данной задачи, с одной стороны, не построен алгоритм решения, имеющий полиномиальную трудоемкость, с другой – не доказана NP-полнота, многие авторы занимаются разработкой и исследованием новых алгоритмов. При этом исследуются разные постановки задачи, в том числе для ориентированных графов [2], однако в большинстве случаев задача рассматривается для неориентированных графов без петель и кратных ребер.

Не так давно Л. Бабай [3] предложил алгоритм решения задачи, имеющий квазиполиномиальную трудоемкость $exp((\log n)^{O(1)})$. Вместе с тем алгоритм весьма сложен для понимания и его корректность, насколько нам известно, на момент написания статьи не была подтверждена. В большинстве случаев авторы при решении задачи используют понятие инварианта [4], то есть некоторой количественной характеристики структуры графа, которая остается неизменной при перенумерации его вершин. В качестве примеров работ, предлагающих алгоритмы решения задачи и их программную реализацию можно назвать [5—10]. В числе прочих предлагаются алгоритмы, которые за полиномиальное время, если и не решают задачу полностью, то вычисляют некоторые характеристики, которые могут быть использованы для решения полной задачи [11, 12].

Отметим, что чаще всего авторы ищут решение задачи в постановке из [1], то есть их алгоритм должен просто ответить на вопрос, изоморфны графы или нет. Обычно решение сводится к попытке построения полных инвариантов обоих графов, сравнение которых и дает ответ на поставленный вопрос. Вместе с тем для практических задач не менее важно построить подстановку, задающую соответствие между вершинами двух графов. Поэтому мы в своей работе будем решать именно эту задачу, тем более, что имея такую подстановку мы сразу можем проверить, действительно ли исходные графы изоморфны. Ввиду высокой трудоемкости решения задачи мы также ставили перед собой цель добиться существенного ускорения за счет построения параллельного алгоритма решения задачи.

В распоряжении автора были программные инструменты для организации параллельных вычислений в соответствии концепцией рекурсивно-параллельного (РП) программирования. Основные принципы организации рекурсивно-параллельных вычислений, и основные алгоритмы и механизмы поддержки этого стиля программирования описаны в [13]. Разработанные автором библиотеки [14, 15] позволяют относительно легко создавать, отлаживать и эксплуатировать РП-приложения в среде .NET Framework. В [16] подробно описаны функциональные возможности упомянутых библиотек. Они успешно применялись при разработке и исследовании параллельных алгоритмов для решения задачи о клике [16], задачи коммивояжера [17] и задачи о рюкзаке [18].

1. Постановка задачи

Напомним формулировку задачи. Пусть есть два неориентированных графа, заданных своими множествами вершин и ребер: $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$. Требуется найти функцию $f(V_1 \longrightarrow V_2)$, такую что $\{u, v\} \in E_1 \iff \{f(u), f(v)\} \in E_2$, либо доказать ее отсутствие. Напомним, что как было отмечено выше, нашей целью является нахождение этой функции, задающей подстановку, или доказательство ее отсутствия.

2. Последовательный алгоритм решения задачи

Сначала опишем предлагаемый последовательный алгоритм, поскольку именно он является основой для дальнейшего распараллеливания решения задачи.

Ключевым понятием при построении алгоритма является инвариант вершины $I(v)$ – характеристика, которая позволяет утверждать, что если $v_1 \in V_1$, $v_2 \in V_2$ и $I(v_1) \neq I(v_2)$, то в искомой подстановке $f(v_1) \neq v_2$. Понятие инварианта вершины очень часто используется для решения задачи об изоморфизме, причем различными авторами предлагаются самые разные варианты их задания [5, 11, 12].

В качестве возможных значений инвариантов вершин при построении своего алгоритма мы рассматривали следующие:

1. Массив $L(v) = \{l_i\}$, $i \in \{1, ..., d\}$, где d – диаметр компоненты графа, к которой принадлежит v, l_i – количество вершин, отстоящих от v на расстояние i.
2. Множество из двух массивов $\{L(v), M(v)\}$, где $L(v)$ имеет тот же смысл, что и в предыдущем варианте, а $M(v) = \{m_i\}$, $i \in \{1, ..., d\}$, где m_i – количество ребер, связывающих между собой вершины, отстоящие от v на расстояние i.
3. Множество из трех массивов $\{L(v), M(v), N(v)\}$. Здесь $L(v)$ и $M(v)$ имеют тот же смысл, что и в предыдущем варианте, а $N(v) = \{n_i\}$, $i \in \{1, ..., d-1\}$, где n_i – количество ребер, связывающих между собой вершины, отстоящие от v на расстояние i с вершинами, находящимися на расстоянии $i + 1$ от вершины v.

Вычисление инвариантов для всех вершин обоих графов было первым шагом нашего алгоритма. Трудоемкость этого этапа, очевидно, оценивается как $O(n^3)$. Отметим сразу, что в ходе эксперимента мы отдали предпочтение второму варианту, он требовал по сравнению с первым в полтора-два раза больше времени, но очень сильно сокращал последующие вычисления. Третий вариант, казалось бы, позволяет более детально характеризовать вершины, однако по сравнению со вторым существенного ускорения на последующих этапах решения задачи не обеспечивал.

Перейдем к описанию базового последовательного алгоритма. При этом мы будем использовать понятия частичной подстановки и перестановки.

Частичная подстановка S представляет собой массив длины n, в котором часть элементов имеет неопределенное значение (по умолчанию), остальным присвоены значения $S[i] = j$, соответствующие значениям функции $f(i) = j$. Подстановка, заполненная целиком, определяет искомую подстановку. Перестановка T представляет собой массив из k номеров вершин. В процессе перебора подстановок нам требуется перебрать все $k!$ вариантов их расположения. Матрицы смежности для первого и второго графа мы обозначим A_1 и A_2.

Предлагаемый алгоритм последовательно выполняет следующие действия:

1. Создаем массивы инвариантов всех вершин обоих графов. При этом одновременно вычисляются диаметры графов (для статистики) и соответственно определяется их связность.
2. Если графы оказались несвязными, разбиваем их на компоненты и дополняем инварианты для каждой вершины указанием на то, к какой компоненте связности она принадлежит. Номера компонент связности для разных графов совпадать не обязаны, они нужны только для того, чтобы после сортировки вершин и разбиения их по группам в каждой группе присутствовали вершины из одной компоненты связности.
3. Сортируем массивы инвариантов вершин для каждого из графов. Отношение порядка между двумя инвариантами задается самым естественным образом – лексикографически.
4. Определяем группы вершин с одинаковыми инвариантами и их размеры. В результате проведенной ранее сортировки группы вершин, потенциально соответствующих друг другу, располагаются в одинаковом порядке для обоих графов. В рамках одной группы, конечно, вершины могут находиться в произвольном порядке.

5. Проверяем на совпадение размеров групп. Если они не совпадают, то графы, очевидно, не изоморфны.
6. Пытаемся максимально измельчить группы. Для этого строим и проверяем частичную подстановку S, которая задает значения искомой подстановки f (пока еще не для всех вершин). В нее включаем все вершины из групп единичного размера – их соответствие друг другу в искомой подстановке (если графы изоморфны) не вызывает сомнений. Каждую из вновь добавленных вершин i проверяем на выполнение равенства $A_2[S[i], S[j]] = A_1[i, j]$, которое должно выполняться для всех вершин j, включенных в частичную перестановку на этом и более ранних этапах.
7. Далее в цикле пробуем сделать группы с одинаковыми инвариантами еще мельче с учетом текущей частичной подстановки S. Для этого вершины в пределах одной группы (они имеют одинаковые инварианты) сортируются (лексикографически) по значению вектора из нулей и единиц, которые указывают на наличие ребра между данной вершиной и вершинами, уже включенными в частичную подстановку. Затем происходит дробление ранее созданных групп на более мелкие с использованием того же способа сравнения. После очередного уменьшения размеров групп выделяем вновь появившиеся группы единичного размера, проверяем измельченные группы на совпадение размеров и в случае совпадения достраиваем текущую частичную подстановку за счет новых групп единичного размера. Цикл завершается, если в результате очередной итерации не появилось новых групп единичного размера.
8. Окончательно решаем задачу перебором оставшихся вариантов подстановок (рекурсивно). Естественно, подстановки перебираются только для идентичных групп вершин в обоих графах, это многократно уменьшает количество вариантов для рассмотрения. Блок-схема алгоритма этого этапа вычислений приводится ниже.

Рассмотрим алгоритм перебора оставшихся вариантов подстановок. Строго говоря, если осталось m групп размера $k_1, k_2, ..., k_m$, количество подстановок, который потребуется перебрать, равно $K = k_1! * k_2! * ... * k_m!$, однако предлагаемый алгоритм на каждой ветке отсекает огромное количество неподходящих подстановок, в результате вычисления заканчиваются очень быстро. В ходе экспериментов, даже если изначально величина K имела порядок $10^{50} – 10^{100}$, на деле требовалось перебрать несколько тысяч или десятков тысяч подстановок, что на современных компьютерах выполняется очень быстро. Зачастую эта величина измерялась десятками или даже единицами.

Блок-схема рекурсивной функции $RecursiveBruteForce(rL)$, используемой для окончательного перебора подстановок приведена на рисунке 1. В ней требуют пояснений некоторые действия, записанные в третьем блоке. Основной причиной радикального замедления работы может быть ситуация, когда в одной группе вершин (напомним, что они принадлежат одной компоненте связности) все вершины имеют одинаковые инварианты, например, образуют кольцо. Они, конечно, могут быть пронумерованы в произвольном порядке, но их требуется сопоставить вершинам идентичной группы во втором графе. А для этого их требуется выстроить по порядку следования в кольце, чтобы не перебирать все варианты, которых может быть чрезвычайно много. Необходимость включения этого блока в алгоритм выяснилась при тестировании программы на регулярных графах со степенью вершины 2. Разумеется, могут встретиться и другие особенные случаи, например, компоненты, представляющие собой полные двудольные графы. Впрочем, в этом случае можно просто инвертировать граф и получить два кольца. Разумеется, такого рода примеров можно построить много, но все они будут носить искусственный характер, предусмотреть все возможные варианты невозможно. Кольца же нередко могут встретиться в обычных ситуациях, если графы, с которыми мы имеем дело, имеют низкую плотность.

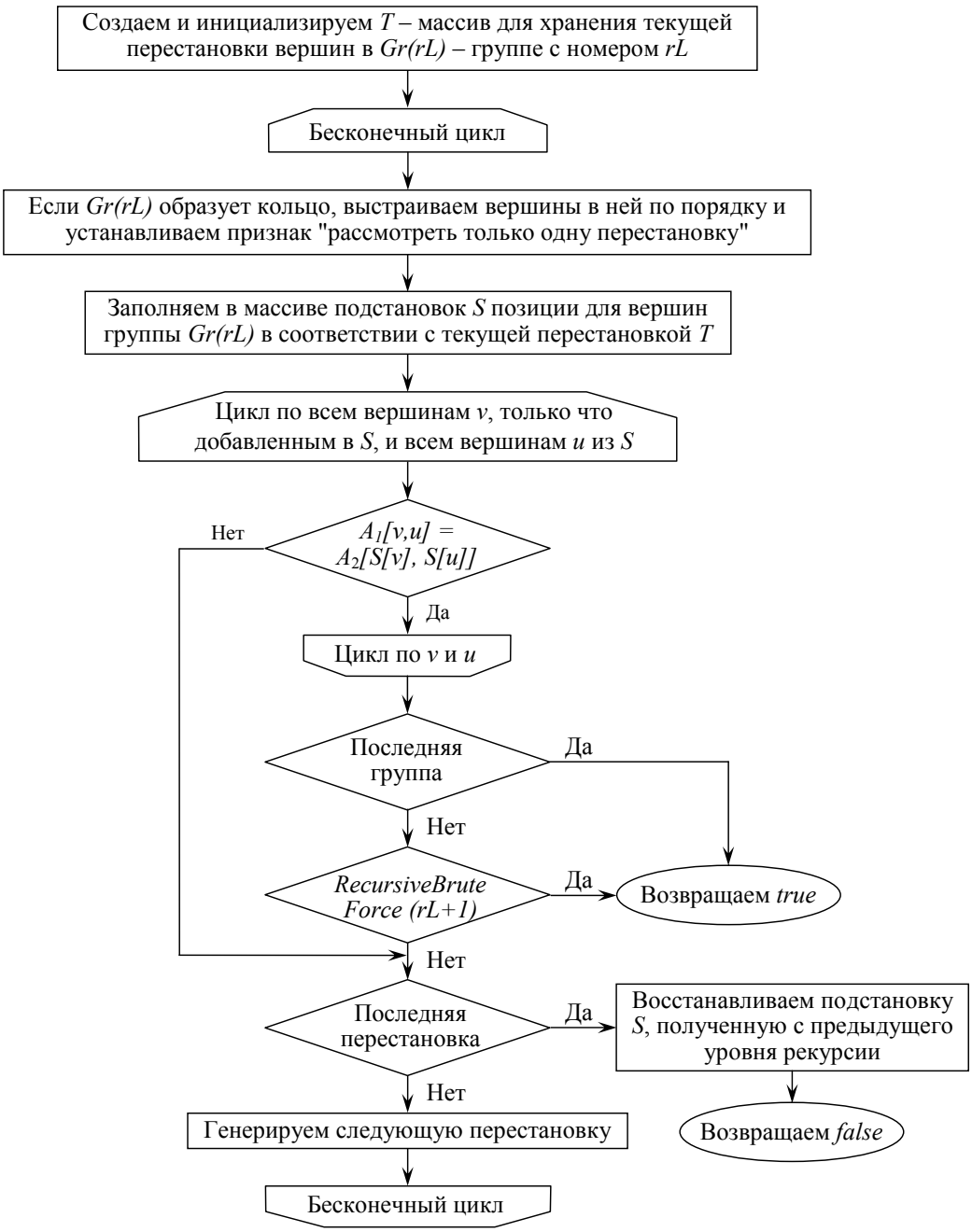


Fig. 1. Block diagram of the function *RecursiveBruteForce(rL)* to enumerate substitutions

Рис. 1. Блок-схема функции перебора подстановок *RecursiveBruteForce(rL)*

3. Распараллеливание алгоритма

В описанном выше последовательном алгоритме основными претендентами на распараллеливание являются первый и последний шаги, поскольку трудоемкость остальных, очевидно, намного ниже, и организация параллельных вычислений для них существенного выигрыша дать не может.

Для первого этапа (вычисление инвариантов для всех вершин обоих графов) распараллеливание легко может быть осуществлено в соответствии со стандартной РП-схемой [13]. Основным параметром каждой задачи является количество вершин, для которых нужно вычислить инварианты. Это количество делится пополам, подзадача для вычисления одной половины остается для решения на данном процессорном модуле (ПМ), другая – оформляется как потенциально мигрирующий процесс и, если необходимо, может быть передана для исполнения на другом ПМ. Библиотека поддержки рекурсивно-параллельного стиля программирования [16] обеспечивает достаточно равномерное распределение работы по системе на начальном этапе вычислений и при необходимости динамическое его перераспределение на последующих этапах.

Автор изначально рассматривал возможность распараллеливания последнего шага алгоритма (окончательный перебор подстановок), однако затем отказался от этой идеи, хотя построение такого РП-алгоритма не представляло сложности. Причина в том, что в ходе экспериментов выяснилось, что несмотря на отсутствие полиномиальной оценки сложности, этот этап не требовал слишком большого количества вычислений. Качество разбиения вершин на группы и эффективность отсечения неперспективных вариантов описанного выше алгоритма во всех экспериментах приводили к очень быстрому получению искомого результата. При таких условиях организация параллельных вычислений могла привести только к замедлению работы из-за неизбежных накладных расходов на порождение и запуск параллельных активаций процедур.

4. Описание эксперимента и его результаты

Исходные данные для тестирования генерировались случайным образом. Для усложнения задачи было принято решение исследовать работу алгоритма на регулярных графах. Было сгенерировано несколько десятков графов с количеством вершин 7000 и степенью вершины от 2 до 3000. Графы с плотностью более 0.5 не использовались для тестирования, поскольку их можно просто инвертировать и исследовать получившиеся графы. Изначально для генерации случайных регулярных графов использовался алгоритм, предложенный в [19], однако для такого количества вершин он работал слишком медленно, и автором был разработан свой, существенно более быстрый алгоритм.

В процессе тестирования использовались компьютеры на базе двухядерного процессора Intel Core i3-7100 (максимальное количество потоков 4) с тактовой частотой 3.90 GHz и 8 GB оперативной памяти, работающие под управлением 64-разрядной ОС Windows 10. Пропускная способность сети равнялась 100 Mb/s.

В таблице 1 приведены некоторые результаты вычислений, позволяющие оценить время решения задачи последовательным алгоритмом и долю вычислений, приходящихся на первый этап (вычисление инвариантов вершин). Мы показали результаты не для всех рассмотренных вариантов исходных данных, а только небольшую их часть, однако они все похожи друг на друга, слегка выделяются только результаты тестирования для графов со степенью вершины, равной 2, они по вполне понятным причинам оказались несвязными.

Для этих же графов мы провели эксперимент по оценке эффективности параллельного алгоритма. В таблице 2 приведены результаты, демонстрирующие ускорение параллельного алгоритма по отношению к последовательному для количества задействованных компьютеров (ПМ) до 16. Показано как ускорение собственно вычислений, так и ускорение с учетом необходимости передачи по сети исходных данных и вычисленных результатов.

Table 1. Execution time of sequential algorithm for regular graphs on 7000 vertices

Таблица 1. Длительность работы последовательного алгоритма для регулярных графов на 7000 вершин

Степень графа	**2**	**3**	**5**	**20**	**200**	**300**
Диаметр графа	∞	16	9	4	3	2
Все вычисления (с)	*941.11*	*3149.81*	*3240.97*	*2381.29*	*2794.24*	*2243.00*
Выч. инвариантов	932.41	3149.22	3240.36	2380.51	2792.84	2241.73
Инварианты, %	*99.08*	*99.98*	*99.98*	*99.97*	*99.95*	*99.94*

Table 2. Speed increase for regular graphs on 7000 vertices

Таблица 2. Ускорение вычислений для регулярных графов на 7000 вершин

Кол-во ПМ	Степень графа	***2***	***3***	***5***	***20***	***200***	***3000***
	Диаметр графа	∞	16	9	4	3	2
1	Уск. вычислений	2.17	2.41	2.38	2.47	2.63	2.84
	Уск. всей работы	*2.17*	*2.41*	*2.38*	*2.47*	*2.63*	*2.84*
2	Уск. вычислений	4.27	4.80	4.69	4.89	5.25	5.59
	Уск. всей работы	*4.13*	*4.76*	*4.66*	*4.85*	*5.20*	*5.52*
4	Уск. вычислений	8.03	9.47	9.37	9.61	9.88	10.81
	Уск. всей работы	*7.59*	*9.34*	*9.24*	*9.43*	*9.72*	*10.53*
6	Уск. вычислений	11.15	13.57	13.47	13.47	14.34	15.39
	Уск. всей работы	*10.33*	*13.29*	*13.20*	*13.11*	*14.01*	*14.82*
8	Уск. вычислений	12.75	15.91	15.43	16.16	17.07	18.61
	Уск. всей работы	*11.34*	*15.33*	*14.86*	*15.43*	*16.32*	*17.39*
12	Уск. вычислений	19.29	25.28	24.83	24.55	26.28	27.66
	Уск. всей работы	*16.80*	*24.23*	*23.87*	*23.26*	*25.05*	*25.69*
16	Уск. вычислений	22.20	31.67	30.55	32.64	33.08	34.96
	Уск. всей работы	*16.81*	*27.91*	*27.38*	*28.21*	*28.94*	*30.05*

Результаты наглядно показывают хорошие перспективы для ускорения алгоритма за счет его распараллеливания. Тот факт, что даже на одном компьютере параллельная версия программы работает быстрее, чем последовательная, объясняется тем, что она задействует для вычислений несколько потоков, что дает ощутимое ускорение на многоядерном процессоре.

References

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co, San Francisco, 1979.

[2] D. C. Schmidt and L. E. Druffel, "A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices", *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 433–445, 1976.

[3] L. Babai, "Graph isomorphism in quasipolynomial time", in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 2016, pp. 684–697.

[4] F. Harary, *Graph theory*. Addison-Wesley, 1969.

[5] Y. German, O. German, and A. Dunaev, "An algorithm for establishing graph's isomorfism", *Proceedings of BSTU. Issue 3, Physics and mathematics. Informatics*, no. 2 (200), pp. 114–117, 2017.

[6] V. K. Pogrebnoy and A. Pogrebnoy, "Polynomial algorithm of computing complete graph invariant on the basis of integral structure descriptor", *Bulletin of the Tomsk Polytechnic University*, vol. 323, no. 5, pp. 152–159, 2013.

[7] V. K. Pogrebnoy and A. Pogrebnoy, "Polynomiality of method for computing graph structure integral descriptor", *Bulletin of the Tomsk Polytechnic University*, vol. 323, no. 5, pp. 146–151, 2013.

[8] A. Pogrebnoy, "Complete graph invariant and algorithm of its computation", *Bulletin of the Tomsk Polytechnic University*, vol. 325, no. 5, pp. 110–122, 2014.

[9] A. Pogrebnoy and V. K. Pogrebnoy, "Method of graph vertices differentiation and solution of the isomorphism problem", *Bulletin of the Tomsk Polytechnic University*, vol. 326, no. 6, pp. 34–45, 2015.

[10] A. Pogrebnoy and V. K. Pogrebnoy, "Method of graph vertices differentiation and solution of the isomorphism problem in geoinformatics", *Bulletin of the Tomsk Polytechnic University*, vol. 326, no. 11, pp. 56–66, 2015.

[11] B. F. Melnikov and N. P. Churikova, "Algorithms of Comparative Analysis of Two Invariants of a Graph", *Sovremennye informacionnye tehnologii i IT-obrazovanie (Modern Information Technologies and IT-Education)*, vol. 15, no. 1, pp. 45–51, 2019.

[12] G. S. Ivanova and V. A. Ovchinnikov, "Completely described undirected graph structure", *Science and Education of the Bauman MSTU*, no. 4, pp. 106–123, 2016.

[13] V. V. Vasilchikov, *Sredstva parallelnogo programmirovaniya dlya vychislitelnykh sistem s dinamicheskoy balansirovkoy zagruzki*. YarGU, Yaroslavl, 2001.

[14] V. V. Vasilchikov, "Kommunikatsionnyy modul dlya organizatsii polnosvyaznogo soedineniya kompyuterov v lokalnoy seti s ispolzovaniem .NET Framework", *Svidetelstvo o gosudarstvennoy registratsii programmy dlya EVM № 2013619925*, 2013.

[15] V. V. Vasilchikov, "Biblioteka podderzhki rekursivno-parallelnogo programmirovaniya dlya .NET Framework", *Svidetelstvo o gosudarstvennoy registratsii programmy dlya EVM № 2013619926*, 2013.

[16] V. V. Vasilchikov, "On the recursive-parallel programming for the. NET framework", *Automatic Control and Computer Sciences*, vol. 48, no. 7, pp. 575–580, 2014.

[17] V. V. Vasilchikov, "On optimization and parallelization of the little algorithm for solving the travelling salesman problem", *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 551–557, 2017.

[18] V. V. Vasilchikov, "On a recursive-parallel algorithm for solving the knapsack problem", *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 810–816, 2018.

[19] A. Steger and N. Wormald, "Generating random regular graphs quickly", *Combinatorics, Probability and Computing*, vol. 8, no. 4, pp. 377–396, 1999.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

The Determination of Distances between Images by de Rham Currents Method

S. N. Chukanov[1]

DOI: 10.18255/1818-1015-2020-1-96-107

[1]Sobolev Institute of Mathematics, SB RAS, Omsk branch, 13 Pevtsova str., Omsk 644043, Russia.

MSC2020: 68U10
Research article
Full text in English

Received February 1, 2020
After revision February 27, 2020
Accepted February 28, 2020

The goal of the paper is to develop an algorithm for matching the shapes of images of objects based on the geometric method of de Rham currents and preliminary affine transformation of the source image shape. In the formation of the matching algorithm, the problems of ensuring invariance to geometric image transformations and ensuring the absence of a bijective correspondence requirement between images segments were solved. The algorithm of shapes matching based on the current method is resistant to changes of the topology of object shapes and reparametrization. When analyzing the data structures of an object, not only the geometric form is important, but also the signals associated with this form by functional dependence. To take these signals into account, it is proposed to expand de Rham currents with an additional component corresponding to the signal structure. To improve the accuracy of shapes matching of the source and terminal images we determine the functional on the basis of the formation of a squared distance between the shapes of the source and terminal images modeled by de Rham currents. The original image is subjected to preliminary affine transformation to minimize the squared distance between the deformed and terminal images.

Keywords: pattern recognition; image matching; de Rham current; affine transformations

INFORMATION ABOUT THE AUTHORS

Sergey N. Chukanov | orcid.org/0000-0002-8106-9813. E-mail: ch_sn@mail.ru
Doctor of Technical Science, Professor.

Funding: This work was supported by the Russian Foundation for Basic Research, projects № 18–07–00526 and № 18–08–01284. This work was supported by the Basic Research Program of the Siberian Branch of the Russian Academy of Sciences № I.5.1., Project № 0314-2019-0020.

For citation: S. N. Chukanov, "The Determination of Distances between Images by de Rham Currents Method", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 96-107, 2020.

© Chukanov S. N., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

Определение расстояний между изображениями методом потоков де Рама

С. Н. Чуканов[1]

DOI: 10.18255/1818-1015-2020-1-96-107

[1]Институт математики им. С. Л. Соболева СО РАН, Омский филиал, ул. Певцова, 13, г. Омск, 644043, Россия.

УДК 004.932.2
Научная статья
Полный текст на английском языке

Получена 1 февраля 2020 г.
После доработки 27 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

Целью работы является разработка алгоритма сравнения форм изображений объектов, основанного на геометрическом методе потоков де Рама и предварительном аффинном преобразовании исходной формы изображения. При формировании алгоритма сравнения решены задачи обеспечения инвариантности к геометрическим преобразованиям изображений и обеспечения отсутствия требования биективного соответствия между сегментами исходного и терминального изображений. Алгоритм сравнения форм, основанный на методе потоков, устойчив к изменению топологии форм объектов и репараметризации. При анализе структур данных объекта имеет значение не только геометрическая форма, но и сигналы, ассоциированные с этой формой функциональной зависимостью. Для учета этих сигналов предлагается расширить потоки де Рама дополнительным компонентом, соответствующим структуре сигнала. Для повышения точности сравнения форм исходного и терминального изображений определяется функционал на основе формирования квадрата расстояния между формами исходного и терминального изображений, моделируемыми потоками де Рама. Исходное изображение подвергается предварительному аффинному преобразованию для минимизации квадрата расстояния между деформированным и терминальным изображениями.

Ключевые слова: распознавание образов; сравнение изображений; поток де Рама; аффинные преобразования

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сергей Николаевич Чуканов | orcid.org/0000-0002-8106-9813. E-mail: ch_sn@mail.ru
д-р техн. наук, профессор.

Финансирование: Работа выполнена при финансовой поддержке РФФИ, проекты № 18–07–00526 и № 18–08–01284. Работа выполнена при поддержке программы фундаментальных научных исследований СО РАН № I.5.1., проект № 0314-2019-0020.

Для цитирования: S. N. Chukanov, “The Determination of Distances between Images by de Rham Currents Method”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 96-107, 2020.

© Чуканов С. Н., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Introduction

Analysis and matching of image shapes of objects is an important problem in pattern recognition [1], image registration [2], biometrics [3], computational anatomy [4]. The determination of distances for matching the shapes of objects is one of the methods for analyzing shapes in pattern recognition. Known distances used in pattern recognition are: Hausdorff, Frechet, Procrustes, Wasserstein and others [5]. One of the most effective methods for matching the shapes of objects is the LDDMM method (Large deformation diffeomorphic metric mapping [6]), in which the distance between the shapes is determined by the minimized functional consisting of the integral of the deformation energy of the original image and the terminal and the sum squared of deviations between the resulting deformable and terminal image.

The traditional methods of matching image shapes in pattern recognition problems have the following disadvantages. Firstly, the lack of invariance of methods in affine transformations of the shapes of images of objects; secondly, the requirement of bijective correspondence between image segments; thirdly, the lack of accounting of the orientation of the shapes of the source and terminal images; fourthly, the lack of accounting of the functional dependence of image segments.

1. Problem statement

Purpose of this paper is to develop an algorithm for matching the image shapes of objects, which is devoid of the above disadvantages. An algorithm for matching shapes based on the geometric de Rham current method [7] and preliminary affine transformation of the original image form is proposed. The method of currents can be used to represent and analyze forms of various nature: point landmarks, curves, surfaces, signals. If $\Omega^k(M)$ is the space of continuous differential k-forms ω in $M \in \mathbb{R}^d$, then the space of de Rham k-currents $\left(\Omega^k\right)^*(M)$ is the dual space to the space $\Omega^k(M)$; k-current $T(\cdot) \in \left(\Omega^k\right)^*(M)$ is a linear functional mapping a differential k-form $\omega \in \Omega^k(M)$: $\omega \longrightarrow T(\omega) \in \mathbb{R}$. For any hypersurface $S \in \mathbb{R}^k$ we can associate such current $T_S(\cdot) \in \left(\Omega^k\right)^*$ that [7]:

$$T_S(\omega) \in \int_S \omega \in; \forall \omega \in \Omega^k.$$

In the formation of the matching algorithm, the following problems were solved: ensuring the invariance to geometric image transformations, ensuring the absence of a bijective correspondence requirement between image segments [8–10]. Using the de Rham current algorithm allow us to increase the accuracy of matching by taking into account the orientation of the segments of the image shape. The algorithm for matching shapes based on the current method is stable when changing the topology of the shapes of objects and changing parameterization.

The problem of correctly determination the distance between currents that decode the shapes of objects is solved by imbedding the space of de Rham currents in RKHS (reproducing kernel Hilbert spaces) [11]. The study of the shapes of objects is proposed to be carried out by forming test vector fields. Since the de Rham current is not a scalar, for working with currents it is necessary to use vector-valued RKHS [12, 13].

When analyzing the data structures of an object, not only the geometric shape is important, but also the signals associated with this shape with functional dependence. Signals can include structures that are more complicated than real numbers; e.g. vector, tensor signals, quaternions, etc. To take these signals into account, it is proposed to expand de Rham currents with an additional component corresponding to the signal structure.

The results of a diffeomorphic matching of the shapes of objects with an extension of the LDDMM algorithm to the case of metamorphosis, in which there may be no bijective correspondence between the segments of the source and final images, are presented in the article [14]. In this case, a functional is formed that corresponds to the image deformation and determines the distance between the shapes of the initial and terminal images. In order to increase the accuracy of matching the shapes of the source and terminal images in this paper, we determine the functional on the basis of the formation of the squared distance

between the shapes of the source and terminal images modeled by de Rham currents. The source image undergoes a preliminary affine transformation formalized by Lie groups to minimize the squared distance between the two shapes. The minimization of the functional of the squared distance between the image shapes constructed using de Rham currents is based on the QPSO algorithm.

2. Hamiltonian mechanics of image points

Representation of an image after a diffeomorphic transformation can be considered as an evolution of point landmarks of an image based on the principles of Hamilton mechanics. Consider the parameterization of the image by particles. Let $q_i(t);\ \ i = 1, \dots, N$ be the position vector of the particle i and $p_i(t); i = 1, \dots, N$ be the corresponding momentum vector in time t. If we assume that the moments and velocities of particles are interconnected by the relation: $p_i = \mathcal{L} \cdot v_i$, where $\mathcal{L}$ is an invertible linear operator, then the inverse operator $\mathcal{L}^{-1}$: $v_i = \mathcal{L}^{-1} \cdot p_i = \mathcal{K} p_i$. For an operator $\mathcal{L} = \mathrm{id} - \alpha\nabla^2$ in space $\mathbb{R}^2$, the inverse operator $\mathcal{K} = \mathcal{L}^{-1}$ can be approximated by the Gauss function: $K\left(q_i - q_j\right) = \beta e^{-\alpha^{-2}\left(q_i - q_j\right)^T\left(q_i - q_j\right)}$.

We construct a functional J_0 corresponding to the deformation of the image represented by a set of points:

$$J_0 = \frac{1}{2}\int_0^1 \left\{ \sum_{i,j=1}^{N} p_i^T K\left(q_i - q_j\right) p_j \right\} dt.$$

Minimization J_0 is carried out according to the values of the components of the momentum vectors $p_i, p_j; i, j = 1 \dots N$. The minimization problem for J_0 can be represented as the optimal control problem with the Hamiltonian: $H_0(q,p) = \frac{1}{2}\sum\limits_{i,j=1}^{N} p_i^T K\left(q_i - q_j\right) p_j$. If the Hamiltonian of the system is taken in the form:

$$H(q,p) = H_0(q,p) + \sigma^{-2}\sum_{i=1}^{N}\left(\dot{q}_i - v_i(q)\right)^2,$$

then the Hamilton equations for derivatives $\dot{\mathbf{p}} = \left(\dot{p}_1, \dots, \dot{p}_N\right), \dot{\mathbf{q}} = \left(\dot{q}_1, \dots, \dot{q}_N\right)$ will take the form [14] :

$$\begin{aligned} \dot{p}_i &= -\frac{\partial H}{\partial q_i} = -\sum_{j=1}^{N} p_i^T \nabla_{q_i} K\left(q_i - q_j\right) p_j; \\ \dot{q}_i &= \frac{\partial H}{\partial p_i} = \sum_{j=1}^{N} K\left(q_i - q_j\right) p_j + \sigma^2 p_i. \end{aligned} \tag{1}$$

3. Matching the shapes of objects

The theory of currents was developed by G. de Rham [7]. The denomination "current" is chosen by analogy with electromagnetism. For example, in accordance with the law of induction of M. Faraday, the intensity of the current in the wire loop caused by a change in the magnetic field is proportional to the change in the flux of this magnetic field through the surface bounded by the loop. This means that if you measure the current strength in the wire for all possible changes in the magnetic field, you can get the loop geometry. In the works [15–17] presents the concept of currents for the formation of a measure of the difference between simplicial complexes, which does not imply a bijective correspondence between the structures of objects. The concept of using currents is to study the shape of objects by forming test vector fields.

Let $\Omega^k(M)$ be the space of continuous differential k-forms ω in $M \in \mathbb{R}^d$. The space of k-currents is the dual space to the space of differential k-forms; k-current is a linear functional mapping a differential k-form ω: $\omega \longrightarrow T(\omega) \in \mathbb{R}$. The form $\omega \in \Omega^{n-1}$ can be integrated over a hypersurface S, which is associated with $(n-1)$-current $T_S \in (\Omega^*)^{n-1}$ in such a way that: $T_S(\omega) = \int\limits_S \omega, \forall\omega \in \Omega^{n-1}$. Suppose that a hypersurface $S$is

parameterized by a surface r: $D \subset \mathbb{R}^{n-1} \longrightarrow \mathbb{R}^n$, with $r(D) = S$. Then:

$$T_S(\omega) = \int_S \omega = \int_D \omega(r(x)) \left(r_{x_1} \wedge \ldots \wedge r_{x_{n-1}}\right) dx_1 \ldots dx_{n-1},$$

where $r_{x_i} = \dfrac{\partial r}{\partial x_i}$; $i = 1, \ldots, n-1$.

Consider the case of plane closed curves and compact surfaces. Let $l : L = [a, b] \longrightarrow \mathbb{R}^2$ be a parametrized curve in $\mathbb{R}^2$. We associate with l such a current $T_l(\cdot)$ that when $T_l(\cdot)$ acting on ω we get: $T_l(\omega) = \int_L (\bar{\omega}(l(t)) \cdot \tau(t)) \frac{\partial l(t)}{\partial t} dt$, where $\tau(t)$ is the tangent vector to l at the point t, $\bar{\omega}$ is the vector field in $\mathbb{R}^2$ corresponding ω. Let S be a surface in $\mathbb{R}^3$, with parameterization $r : (u, v) \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^3$; $r(u, v) = S$. We associate with S such a current $T_S(\cdot)$ that when acting $T_S(\cdot)$ on ω we get: $T_S(\omega) = \int_U \bar{\omega}(r(u, v)) \cdot (r_u \times r_u) \, du dv$, where $\bar{\omega}$ is the vector field in $\mathbb{R}^3$, corresponding to ω, "$\times$" is the vector product operator.

Let $(W, \langle \cdot, \cdot \rangle_W)$ be a test Hilbert space of vector fields $\mathbb{R}^n \longrightarrow \mathbb{R}^n$. We introduce W^* – the space of currents dual to the space W, that is, the space of continuous linear mappings: $W \longrightarrow \mathbb{R}$. For any current $T_S(\cdot) \in W^*$, there is such a representation $K^W T_S \in W$ that $T_S(\omega) = \left\langle K^W T_S, \omega \right\rangle_W, \forall \omega \in W$. The space W is a vector-valued RKHS (see Appendix 1), W equipped with an inner product $\left\langle K^W(\cdot, x)\alpha, K^W(\cdot, y)\beta \right\rangle_W = \alpha^T K^W(x, y)\beta$, that is defined for the fields $K^W(\cdot, x)\alpha$ and $K^W(\cdot, y)\beta$. If we denote $K^W(\cdot, y)\beta$ as ω , then we obtain the reproducing property: $\left\langle K^W(\cdot, x)\alpha, \omega \right\rangle_W = \alpha^T \omega(x); \forall \omega \in W$.

There is a linear mapping: $L_W : W \longrightarrow W^*$, between space W and the corresponding space of currents: $W^* : L_W(\omega)\left(\omega'\right) = \left\langle \omega, \omega' \right\rangle_W, \forall \omega, \omega' \in W$. The inner product $\langle \cdot, \cdot \rangle_W$ can be mapped to the current space W^* using linear mapping L_W. Then the inner product is between two currents T, T': $\left\langle T, T' \right\rangle_{W^*} = \left\langle L_W^{-1}(T), L_W^{-1}\left(T'\right) \right\rangle_W$.

In space W, the basic elements are fields of the form $K^W(\cdot, x)\alpha$, and the corresponding basic elements in space W^* are the Dirac δ-currents: $\delta_x^\alpha = L_W^{-1}\left(K^W(\cdot, x)\alpha\right)$. From the definition δ_x^α and L_W we get: $\delta_x^\alpha(\omega) = \left\langle K^W(\cdot, x)\alpha, \omega \right\rangle_W = \alpha^T \omega(x)$. Inner product between Dirac δ-currents:

$$\left\langle \delta_x^\alpha, \delta_y^\beta \right\rangle_{W^*} = \langle K(\cdot, x)\alpha, K(\cdot, y)\beta \rangle_W = \alpha^T K^W(x, y)\beta.$$

If the current T represents a curve (or surface), then it can be decomposed into many tangents (normals). The dual representation $\mathcal{L}_W^{-1}(T)$ of the current (vector field in W) is the convolution of all tangents (normals) with the kernel K^W. Polygons of the curve (surface mesh) can be approximated by a finite sum: $T \sim \sum_k \delta_{x_k}^{\alpha_k}$, where x_k is the center of each segment (mesh cell) and α_k is the tangent (normal to the surface) at the point x_k. The value α_k encodes the size of the segment (surface mesh). The dual representation of the current at any point x is given by the sum: $\sum_k K^W(x, x_k)\alpha_k$. The integrals of currents in the discrete approximation are replaced by the sums for the curves: $T_l(\omega) \sim \sum_k \omega(x_k)^T \tau_k$, where τ_k is the tangent at a point x_k; for surfaces: $T_S(\omega) \sim \sum_k \omega(x_k)^T n_k$, where n_k is the normal to the surface at a point x_k.

4. The distance between the shapes of objects

The inner product between two sets of Dirac currents: $T = \sum_i \delta_{x_i}^{\alpha_i}$, $T' = \sum_j \delta_{y_j}^{\beta_j}$, can be determined from the relation: $\left\langle T, T' \right\rangle_{W^*} = T L_W^{-1} T' = \sum_i \sum_j \alpha_i^T K^W\left(x_i, y_j\right)\beta_j$.

We define the square of the distance between two shapes simulated by currents:

$$\begin{aligned} &d\left(T,T'\right)^2 = \left\|T-T'\right\|^2_{W^*} = \left(T-T'\right)L_W^{-1}\left(T-T'\right) = \\ &= \sum_{p=1}^{N}\sum_{q=1}^{N}\alpha_{xp}^T K^W\left(x_p,x_q\right)\alpha_{xq} - \\ &-2\sum_{p=1}^{N}\sum_{q=1}^{N}\alpha_{xp}^T K^W\left(x_p,y_q\right)\alpha_{yq} + \sum_{p=1}^{N}\sum_{q=1}^{N}\alpha_{yp}^T K^W\left(y_p,y_q\right)\alpha_{yq}, \end{aligned} \tag{2}$$

where $K^W\left(x_p,x_q\right) = \exp\left(-\left\|x_p-x_q\right\|^2\lambda_W^{-2}\right)$. To take into account the diffeomorphic deformation of the source shape, it is necessary to add the functional J_0 multiplied by the regularization coefficient to the squared distance $d\left(T,T'\right)^2$.

If the curve l is given by simplicial complexes with points $(x_1,y_1),\ldots,(x_N,y_N),(x_{N+1},y_{N+1})$, then the centers of the segments between adjacent points of the corresponding complexes: $c_{xi} = \frac{(x_i+x_{i+1})}{2}$, $c_{yi} = \frac{(y_i+y_{i+1})}{2}$, and the tangents formed by these segments: $\alpha_{xi} = \frac{(x_{i+1}-x_i)}{2}$, $\alpha_{yi} = \frac{(y_{i+1}-y_i)}{2}$; $i = 1,2,\ldots,N$. Then: $l \longrightarrow T_l(\omega)\sum_{j=1}^{N} K\left(c_j,\cdot\right)\left(\alpha_j\right)$. If S is an oriented triangulated surface defined by points: $(x_1,y_1,z_1),\ldots,(x_N,y_N,z_N)$, $(x_{N+1},y_{N+1},z_{N+1})$, where each j-th triangle is represented by the center: $c_{x_j} = \frac{\left(x_j+x_{j+1}+x_{j+2}\right)}{3}$, $c_{y_j} = \frac{\left(y_j+y_{j+1}+y_{j+2}\right)}{3}$, $c_{z_j} = \frac{\left(z_j+z_{j+1}+z_{j+2}\right)}{3}$, and by a normal vector n_j to the j-th triangle, whose norm encodes the area of the triangle. Then: $S \longrightarrow T_S(\omega)\sum_{j=1}^{N} K\left(x_j,\cdot\right)\left(n_j\right)$.

If the set $\left(x_p,\alpha_p\right)_{p=1\ldots N}$ contains functions f_{x_p} representing signals at the points x_p: $\left(x_p,\alpha_p,f_{x_p}\right)_{p=1\ldots N}$, then the square of the distance $\left\|T-T'\right\|^2_{W^*}$ in (2) can be represented as:

$$\begin{aligned} &d\left(T,T'\right)^2 = \left\|T-T'\right\|^2_{W^*} = \sum_{p=1}^{N}\sum_{q=1}^{N}K^f\left(f_{x_p},f_{x_q}\right)\cdot\alpha_{xp}^T K^W\left(x_p,x_q\right)\alpha_{xq} - \\ &-2\sum_{p=1}^{N}\sum_{q=1}^{N}K^f\left(f_{x_p},f_{y_q}\right)\cdot\alpha_{xp}^T K^W\left(x_p,y_q\right)\alpha_{yq} + \\ &+\sum_{p=1}^{N}\sum_{q=1}^{N}K^f\left(f_{y_p},f_{y_q}\right)\cdot\alpha_{yp}^T K^W\left(y_p,y_q\right)\alpha_{yq}, \end{aligned} \tag{3}$$

where: $K^f\left(f_{x_p},f_{x_q}\right) = \exp\left(-\left(f_{x_p}-f_{x_p}\right)^2\lambda_f^{-2}\right)$, λ_f is the standard deviation f_{x_p} in the space of functions.

4.1. Example 1

Consider an example of matching the shapes of objects. Let a simplicial complex with a set of points $x_1,\ldots,x_n$ be given. If the complex is approximated by a curve, then the centers of the segments and the tangents have the form: $c_i = \frac{(x_i+x_{i+1})}{2}$, $\alpha_i = \frac{(x_{i+1}-x_i)}{2}$, respectively.

Let us consider a matching of the shapes of objects: a square T with vertices: $x = \left(\binom{1}{1},\binom{-1}{1},\binom{-1}{-1},\binom{1}{-1}\right)$, centers of edges: $c^x = \left(\binom{0}{1},\binom{-1}{0},\binom{0}{-1},\binom{1}{0}\right)$, covectors corresponding to tangents to edges: $\alpha^x = \left(\binom{-1}{0}^T,\binom{0}{-1}^T,\binom{1}{0}^T,\binom{0}{1}^T\right)$, and a triangle T' with vertices: $y = \left(\binom{0}{1},\binom{-\frac{\sqrt{3}}{2}}{-\frac{1}{2}},\binom{\frac{\sqrt{3}}{2}}{-\frac{1}{2}}\right)$, centers of edges: $c^y = \left(\binom{-\frac{\sqrt{3}}{4}}{\frac{1}{4}},\binom{0}{-\frac{1}{2}},\binom{\frac{\sqrt{3}}{4}}{\frac{1}{4}}\right)$, covectors corresponding to tangents to edges: $\beta^y = \left(\binom{-\frac{\sqrt{3}}{4}}{-\frac{3}{4}}^T,\binom{\frac{\sqrt{3}}{2}}{0}^T,\binom{-\frac{\sqrt{3}}{4}}{\frac{3}{4}}^T\right)$.

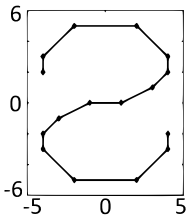
The square of the distance $d\left(T,T'\right)^2 = \left\|T-T'\right\|^2_{W^*}$ with $\lambda_V = 1$, according to (2), is equal to $d\left(T,T'\right)^2 = 1,748$. If there are functions f_{x_p} representing signals at the vertices x_p:

$f_{x_1} = 1, f_{x_2} = 2, f_{x_3} = 3, f_{x_4} = 4$; and the functions f_{y_p} representing the signals in y_p: $f_{y_1} = 1, f_{y_2} = 2, f_{y_3} = 3$, are included in the sets $\left(x_p, \alpha_p\right)_{p=1\ldots4}$ and $\left(y_{p'}, \beta_{p'}\right)_{p'=1\ldots3}$, then the square of the distance $d\left(T, T'\right)^2 = \left\|T - T'\right\|^2_{W^*}$ with $\lambda_f = 1$, according to (3), is equal $d\left(T, T'\right)^2 = 1,966$.

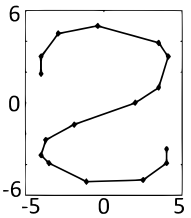
4.2. Example 2

Let us consider an example of a diffeomorphic deformation of the image shape of a symbol of an indefinite shape into an image shape of the shape of number 2 (Fig. 1), number 7 (Fig. 2) and number 8 (Fig. 3).

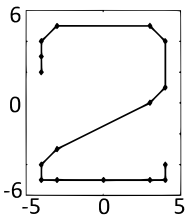
The evolution of deformations of a diffeomorphic shape was determined based on the solution of equations (1). The functional is minimized by values using the QPSO algorithm (see Appendix 2, [18]). In fig. 1, 2, 3 shows intermediate shapes of images for times: $t = 0$ (source image shape), $t = 0,5$ (intermediate image shape), $t = 1$ (terminal image shape).



(a) $t = 0$



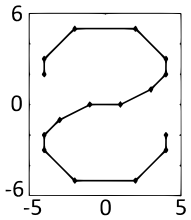
(b) $t = 0,5$



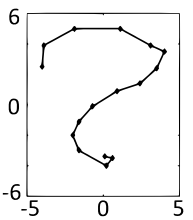
(c) $t = 1$

Fig. 1. Deformation of the shape of the symbol in the shape of number 2

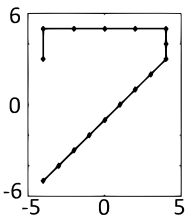
Fig 1. Деформация формы символа в форму цифры 2



(a) $t = 0$



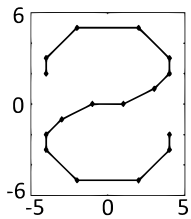
(b) $t = 0,5$



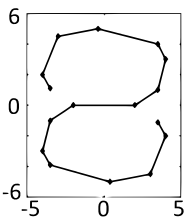
(c) $t = 1$

Fig. 2. Deformation of the shape of the symbol in the shape of number 7

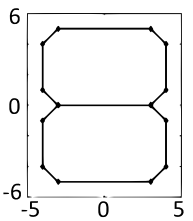
Fig 2. Деформация формы символа в форму цифры 7



(a) $t = 0$



(b) $t = 0,5$



(c) $t = 1$

Fig. 3. Deformation of the shape of the symbol in the shape of the number 8

Fig 3. Деформация формы символа в форму цифры 8

In this case, the values of the squared distance between the source image and the terminal shape $d^2\left(T, T'\right)$, determined from relation (2) with $\lambda_W = 1$, are:

- for the case of deformation of the shape of the symbol in the shape of numbers 2: $d^2\left(T, T'\right) = 78,6$;
- for the case of deformation of the shape of the symbol in the shape of the number 7: $d^2\left(T, T'\right) = 78,0$;
- for the case of deformation of the shape of the symbol in the shape of the figure 8: $d^2\left(T, T'\right) = 16,8$.

Therefore, the algorithm recognizes the character as the number 8.

It should be noted that during deformation of the shape of the symbol into the shape of the figure 8, the topological genus of the shape changes from 0 to 1, that is, the deformation is not a diffeomorphism, but a metamorphosis.

5. Normalization of images based on affine transformations

To improve the accuracy of matching of source and terminal images, these images should be normalized. Below we propose such a normalization method, in which the original image undergoes affine transformation and the functional between the converted original and terminal images is minimized. After that, the normalized original image undergoes a diffeomorphic transformation, while the distance (2) between the converted and terminal images is reduced, which will increase the accuracy of the matching.

An affine transformation is a special case of a diffeomorphic transformation. An affine transformation can be represented in the form [19]:

$$\mathbf{x} \longrightarrow \mathbf{y} = \mathbf{M} \cdot \mathbf{x} + \mathbf{b},$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is an invertible matrix, $\mathbf{b} \in \mathbb{R}^n$, x, y are vectors in an affine space $X \in \mathbb{R}^n$.

In the case of an affine transformation of a curve (surface) point p approximating the shape of a deformable object, it can be represented as: $\mathbf{y}_p \longrightarrow \mathbf{M} \cdot \mathbf{x}_p + \mathbf{b}$, $p = 1, \ldots, P$. As the minimized functional, we choose the square of the distance between the points of the source and final images: $J(\mathbf{M}, \mathbf{b}) = d\left(T, T'\right)^2$, where $d\left(T, T'\right)^2$ it is determined in accordance with (2), T is the current corresponding to the initial shape of the object, T' is the current corresponding to the shape of the deformable object after affine transformation. Let ξ^j be the parameters of the affine transformation: $\xi^j \in \Xi; j = 1, \ldots, N$, where Ξ, is the set of matrix components $\mathbf{M}$ and vector components $\mathbf{b}$.

The values of the parameters ξ_i of the particle i can be found using the QPSO algorithm (quantum particle swarm optimization, see Appendix 2, [18]) to minimize the functional $J(\Xi)$. We denote the value of the minimized functional E_n on the set: $\xi^j_{i,n} \in \Xi$: $E_n = J\left(\xi^j_{1,n}, \ldots, \xi^j_{I,n}\right)$, where n is the iteration step number, and $i \in [1 \ldots I]$ is the particle number. Let $P_{i,n}$ be the values of the parameters that provide the smallest value of the functional E_n for the particle i after the n-th iteration, and G_n be the values of the parameters that provide the smallest value of the functional E_n for all particles after the n-th iteration. We choose the values of the best values of the parameters from the relation:

$$p_{i,n} = \phi_{i,n} \cdot P_{i,n} + (1 - \phi_{i,n}) \cdot G_n,$$

where $\phi_{i,n} \in [0 \ldots 1]$ is a random number of a uniform distribution. The parameters ξ_i of the particle i at the next iteration step $(n + 1)$ can be determined from the relation:

$$\begin{aligned} &\text{if } (\psi_{i,n} < 0,5) \text{ then } \xi^j_{i,n+1} = p^j_{i,n} - \beta \cdot \left|\xi^j_{i,n} - p^j_{i,n}\right| \cdot \ln\left(u^j_{i,n+1}\right); \\ &\text{else } \xi^j_{i,n+1} = p^j_{i,n} + \beta \cdot \left|\xi^j_{i,n} - p^j_{i,n}\right| \cdot \ln\left(u^j_{i,n+1}\right), \end{aligned} \tag{4}$$

where $\psi_{i,n} \in [0 \ldots 1]$, $u^j_{i,n} \in [0 \ldots 1]$ are random numbers of uniform distribution.

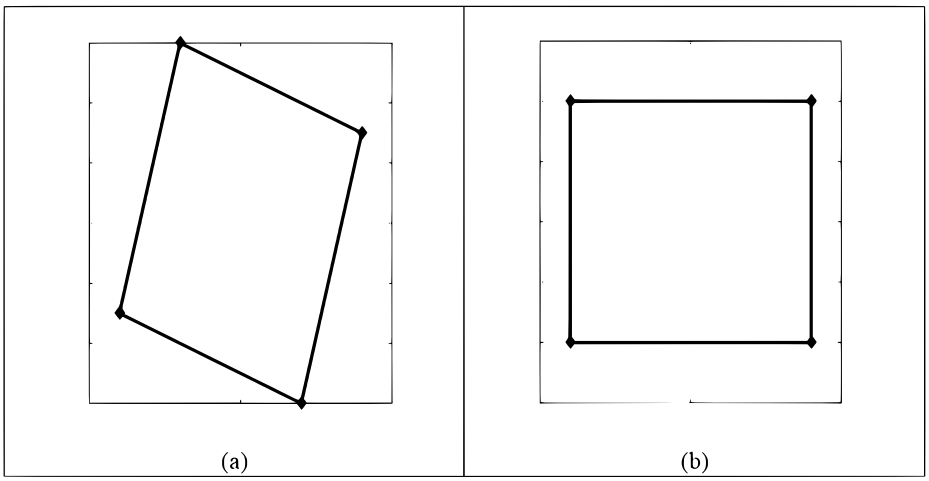


Fig. 4. Example of affine transformation

Fig 4. Пример аффинного преобразования

5.1. Example 3

Consider the example of the affine transformation of a quadrangle T with vertices $\mathbf{x} = \left(\begin{pmatrix}-4\\-3\end{pmatrix} \begin{pmatrix}-2\\6\end{pmatrix} \begin{pmatrix}4\\3\end{pmatrix} \begin{pmatrix}2\\-3\end{pmatrix}\right)$, into a square T' with vertices $\mathbf{x} = \left(\begin{pmatrix}4\\4\end{pmatrix} \begin{pmatrix}4\\-4\end{pmatrix} \begin{pmatrix}-4\\-4\end{pmatrix} \begin{pmatrix}-4\\4\end{pmatrix}\right)$: $\mathbf{x} \longrightarrow \mathbf{y} = \mathbf{M} \cdot \mathbf{x} + \mathbf{b}$; (see fig. 4).

Before the affine transformation, the value $d\left(T, T'\right)$ (see (2)) is equal $d\left(T, T'\right) = 8,2$. After carrying out the affine transformation and minimizing the distance $d\left(T, T'\right)$, we obtain the required components of the matrix $\mathbf{M}$: $\mathbf{M} = \begin{pmatrix}1,2 & -0,26\\0,38 & 0,8\end{pmatrix}$, and the vector $\mathbf{b}$: $\mathbf{b} = \begin{pmatrix}0 & 0\end{pmatrix}^T$. Preliminary affine transformation reduced the distance to $d\left(T, T'\right) = 0,67$.

Conclusion

The paper considered an algorithm for matching image shapes, based on the de Rham currents method and preliminary affine transformation of the source image shape. The de Rham current method can be used to represent shapes of various nature: point landmarks, curves, surfaces, signals. Using the proposed matching algorithm allows us to solve the problem of ensuring invariance to geometric transformations of images and ensuring the absence of a bijective correspondence requirement between image segments. The algorithm for matching shapes based on the current method is stable when changing the topology of the shapes of objects and changing parameterization. An application of the method of reproducing kernel Hilbert space (RKHS) to obtain metrics of the shape of an object is proposed.

To increase the accuracy of matching the shapes of the source and terminal images, it is proposed that the source image be subjected to preliminary affine transformation. The problem of invariance to geometric transformations of images (translation, rotation, scaling, skew) is solved. The minimization of the functional of the squared distance between the image shapes is based on the QPSO algorithm.

The results of a diffeomorphic matching of the shapes of objects with the extension of the LDDMM (large deformation diffeomorphic metric mapping) algorithm to the case of metamorphosis, in which there may be a bijective correspondence between the segments of the source and terminal images, are presented. To improve the accuracy of matching the shapes of the source and terminal images, we determine the functional on the basis of the formation of a squared distance between the shapes of the source and terminal images modeled by de Rham currents.

Appendix 1. Reproducing kernel Hilbert spaces

RKHS (reproducing kernel Hilbert spaces) is a Hilbert space of functions in which a point esti-mation is a continuous linear functional [11]. If two functions in RKHS are close in norm: $\|f-g\| \to 0$, then $|f(x)-g(x)| \to 0; \forall x$. For kernel $k\left(x, x'\right)$, we construct a Hilbert space so that $k\left(x, x'\right)$ is a scalar product in this space. For given points $x_1, x_2, ..., x_n$, we define the Gram matrix: $K_{ij} = k\left(x_i, x_j\right)$. We say that a kernel is positive definite if its Gram matrix is positive definite for all $x_i, x_j; i, j = 1, \ldots, n$. We define a linear functional L_x in a Hilbert space H that estimates each function at a point x: $L_x : f \to f(x), \forall f \in H$. Space H is generated by the reproducing kernel, if $L_x(f)$ is a continuous function for all $x \in X$. The estimation of functional L_x can be represented by taking the inner product of the function f with the function of the reproducing kernel $k(\cdot, x) \in H$. Define a map $\Phi : x \to k(\cdot, x)$. i.e. with each point x in the source space we associate a function $k(\cdot, x)$ with a reproducing property: $f(x) = L_x(f) = \langle f,\ k(\cdot, x)\rangle;\ \forall f \in H, \forall x \in X$. Since $k(\cdot, x) \in H$, then: $k(y, x) = L_y(k(\cdot, x)) = \langle k(\cdot, x), k(\cdot, y)\rangle$, where $k(\cdot, y) \in H$ is the element associated with L_y. This allows us to define the reproducing kernel for H as a function $K : X \times X \to \mathbb{R}$: $k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle$. We construct a vector space RKHS containing all linear combinations of functions $k(\cdot, x) : f(\cdot) = \sum_{j=1}^{m} \alpha_i k(\cdot, x_i)$. Let be: $g(\cdot) = \sum_{j=1}^{m'} \beta_j k\left(\cdot, x_j'\right)$; define the inner product:

$$\langle f, g\rangle = \sum_{i=1}^{m}\sum_{j=1}^{m'} \alpha_i \beta_j k\left(x_i, x_j'\right).$$

For any function: $f(\cdot) = \sum_{j=1}^{m} \alpha_i k(\cdot, x_i)$, the following relation is valid:

$$\langle k(\cdot, x), f\rangle = \sum_{i=1}^{m} \alpha_i k(x_i, x) = f(x).$$

The kernels are analogues of Dirac δ -functions. In space L_2:

$$\langle \delta(\cdot, x), f\rangle = \int f(t)\,\delta(t, x)\,dt = f(x),$$

where $\delta(t, x)$ is the Dirac δ-function.

Appendix 2.
Quantum particle swarm optimization algorithm

The PSO algorithm is presented in [20]. The PSO algorithm considers a set of particles; each particle is a suitable solution to the optimization problem. In terms of classical mechanics, a particle is represented by a vector of its position and a velocity vector, which determine the trajectory of the particle. In quantum mechanics, the term "trajectory" does not make sense, since, in accordance with the principle of uncertainty, the coordinates and velocities of particles cannot be deter-mined simultaneously. A model with a quantum-mechanical potential well based on E. Schrödinger equation [18] is considered below. In quantum mechanics, the state of a particle is deter-mined by the wave function $\psi(x,t)$. In one-dimensional space, the wave function of a particle determines $Q(x,t)$: $|\psi(x,t)|^2\,dx = Q(x,t)\,dx$, where $Q(x,t)\,dx$ is the probability that a measurement of the particle's position at a certain point in time will find it in a neighborhood relative to a point x with the volume of the neighborhood dx. The probability density function satisfies the relation:

$$\int_{-\infty}^{\infty} |\psi|^2\,dx = \int_{-\infty}^{\infty} Q dx = 1.$$

The wave function $\psi(x,t)$ changes in time in accordance with E. Schrödinger equation: $i\hbar\frac{\partial}{\partial t}\psi(x,t) = \hat{H}\psi(x,t)$. For a particle of mass m in a potential field $V(x)$, the Hamilton operator $\hat{H}$ is given by the formula: $\hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + V(x)$, where $\hbar$ is Planck's constant.

Suppose that each particle moves in an δ-potential well in the search space whose center is a point p. The potential energy of a particle in a one-dimensional δ-potential well is represented in the form: $V(x) = -\gamma\cdot\delta(x-p)$. Let be: $y = x - p$. Solving the Schrödinger equation for $y \neq 0$, we obtain the probability density function:

$$Q(y) = |\psi(y)|^2 = L^{-1}\exp\left(-2|y|L^{-1}\right),$$

where L is the characteristic "length" of the δ-potential well. Let s be a uniformly distributed random number: $s = L^{-1}u$; $u = \mathrm{rand}(0,1)$. Replacing $|\psi(y)|^2$ with s, we get: $s = L^{-1}\cdot\exp\left(-2|y|L^{-1}\right)$; $y = x - p = \pm\frac{L}{2}\ln\left(u^{-1}\right)$, consequently: $x = p \pm \dfrac{L}{2}\ln\left(u^{-1}\right)$. We form L at the k-th step of the iteration: $L = \beta\cdot|x_k - p|$, where β is the parameter that controls the search process.

Let $P_{i,n}$ be the values of the parameters that provide the smallest value of the functional E_n for the particle i after the n-th iteration, and G_n be the values of the parameters that provide the smallest value of the functional E_n for all particles after the n-th iteration. We choose the values of the best values of the parameters from the relation: $p_{i,n} = \phi_{i,n}\cdot P_{i,n} + (1-\phi_{i,n})\cdot G_n$, where $\phi_{i,n} \in [0\ldots1]$ is a random number of a uniform distribution. The parameters ξ_i of the particle i at the next iteration step $(n+1)$ can be determined from the relation:

$$\text{if } (\psi_{i,n} < 0,5) \text{ then } \xi^j_{i,n+1} = p^j_{i,n} - \beta\cdot\left|\xi^j_{i,n} - p^j_{i,n}\right|\cdot\ln\left(u^j_{i,n+1}\right);$$
$$\text{else } \xi^j_{i,n+1} = p^j_{i,n} + \beta\cdot\left|\xi^j_{i,n} - p^j_{i,n}\right|\cdot\ln\left(u^j_{i,n+1}\right),$$

where $\psi_{i,n} \in [0\ldots1]$, $u^j_{i,n} \in [0\ldots1]$ are random numbers of uniform distribution.

References

[1] K. Grauman and B. Leibe, "Visual object recognition", *Synthesis lectures on artificial intelligence and machine learning*, vol. 5, no. 2, pp. 1–181, 2011.

[2] A. Goshtasby, *Theory and applications of image registration*. John Wiley & Sons, 2017.

[3] D. Zhang, G. Lu, and L. Zhang, *Advanced biometrics*. Springer, 2018.

[4] M. Miller, A. Trouvé, and L. Younes, "Hamiltonian systems and optimal control in computational anatomy: 100 years since D'Arcy Thompson", *Annual review of biomedical engineering*, vol. 17, pp. 447–509, 2015.

[5] M. Deza and E. Deza, "Encyclopedia of distances", in *Encyclopedia of distances*, Springer-Verlag Berlin Heidelberg, 2016.

[6] L. Younes, *Shapes and diffeomorphisms*. Springer-Verlag Berlin Heidelberg, 2019, vol. 171.

[7] G. De Rham, F. Smith, and S. Chern, *Differentiable manifolds: forms, currents, harmonic forms*. Springer-Verlag, 1984, vol. 266.

[8] S. Chukanov, "A rotation, translation, and scaling invariant Fourier transform of 3D image function", *Optoelectronics, Instrumentation and Data Processing*, vol. 44, no. 3, pp. 249–255, 2008.

[9] S. Chukanov, "Constructing invariants for visualization of vector fields defined by integral curves of dynamic systems", *Optoelectronics, Instrumentation and Data Processing*, vol. 47, no. 2, pp. 151–155, 2011.

[10] S. Chukanov, "Comparison of objects' images based on computational topology methods", *Trudy SPIIRAN*, vol. 18, no. 5, pp. 1043–1065, 2019.

[11] N. Aronszajn, "Theory of reproducing kernels", *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[12] C. Micchelli and M. Pontil, "On learning vector-valued functions", *Neural computation*, vol. 17, no. 1, pp. 177–204, 2005.

[13] J. Glaunes and M. Micheli, "Matrix-valued kernels for shape deformation analysis. Geometry", *Imaging and Computing*, vol. 1, no. 1, pp. 57–139, 2014.

[14] S. Lejhter and S. Chukanov, "Matching of images based on their diffeomorphic mapping", *Computer optics*, vol. 42, no. 1, pp. 96–104, 2018.

[15] S. Barahona, X. Gual-Arnau, M. Ibá nez, and A. Simó, "Unsupervised classification of children?s bodies using currents", *Advances in Data Analysis and Classification*, vol. 12, no. 2, pp. 365–397, 2018.

[16] I. Kaltenmark, B. Charlier, and N. Charon, "A general framework for curve and surface comparison and registration with oriented varifolds", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3346–3355.

[17] M. Vaillant and J. Glaunès, "Surface matching via currents", in *Biennial International Conference on Information Processing in Medical Imaging*, Springer, 2005, pp. 381–392.

[18] D. Tang, Y. Cai, J. Zhao, and Y. Xue, "A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems", *Information Sciences*, vol. 289, pp. 162–189, 2014.

[19] J. Flusser, B. Zitova, and T. Suk, *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, vol. 4, 1995, pp. 1942–1948.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

A Markov Model of Non-Mutually Exclusive Cyber Threats and its Applications for Selecting an Optimal Set of Information Security Remedies

A. A. Kassenov[1], A. A. Magazev[1], V. F. Tsyrulnik[1]

DOI: 10.18255/1818-1015-2020-1-108-123

[1]Omsk State Technical University, 11 Mira pr., Omsk, 644050 Russia.

MSC2020: 68M25
Research article
Full text in Russian

Received October 27, 2019
After revision February 20, 2020
Accepted February 28, 2020

In this work, we study a Markov model of cyber threats that act on a computer system. Within the framework of the model the computer system is considered as a system with failures and recoveries by analogy with models of reliability theory. To estimate functionally-temporal properties of the system we introduce a parameter called the lifetime of the system and defined as the number of transitions of the corresponding Markov chain until the first hit to the final state. Since this random variable plays an important role at evaluating a security level of the computer system, we investigate in detail its random distribution for the case of mutually exclusive cyber threats; in particular, we derive explicit analytical formulae for numerical characteristics of its distribution: expected value and dispersion. Then we generalize substantially the Markov model dropping the assumption that cyber threats acting on the system are mutually exclusive. This modification leads to an extended Markov chain that has (at least qualitatively) the same structure as the original chain. This fact allowed to generalize the above analytical results for the expected value and dispersion of the lifetime to the case of non-mutually exclusive cyber threats. At the end of the work the Markov model for non-mutually exclusive cyber threats is used to state a problem of finding an optimal configuration of security remedies in a given cyber threat space. It is essential that the formulated optimization problems belong to the class of non-linear discrete (Boolean) programming problems. Finally, we consider an example that illustrate the solution of the problem on selecting the optimal set of security remedies for a computer system.

Keywords: cyber threat; Markov chain; security remedy; optimization

INFORMATION ABOUT THE AUTHORS

Adil A. Kassenov	orcid.org/0000-0002-2770-1144. E-mail: kassenov_adil@mail.ru graduate student.
Alexey A. Magazev correspondence author	orcid.org/0000-0002-8725-9183. E-mail: magazev@omgtu.ru doctor of sc., professor.
Valeriya F. Tsyrulnik	orcid.org/0000-0002-6875-7216. E-mail: lera.tsyrulnik@mail.ru postgraduate student.

Funding: The reported study was funded by RFBR, project number 19-37-90122.

For citation: A. A. Kassenov, A. A. Magazev, and V. F. Tsyrulnik, “A Markov Model of Non-Mutually Exclusive Cyber Threats and its Applications for Selecting an Optimal Set of Information Security Remedies”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 108-123, 2020.

© Kassenov A. A., Magazev A. A., Tsyrulnik V. F., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

Марковская модель совместных киберугроз и ее применение для выбора оптимального набора средств защиты информации

А. А. Касенов[1], А. А. Магазев[1], В. Ф. Цырульник[1]

DOI: 10.18255/1818-1015-2020-1-108-123

[1]Омский государственный технический университет, пр. Мира, 11, Омск, 644050 Россия.

УДК 51-74, 004.942
Научная статья
Полный текст на русском языке

Получена 27 октября 2019 г.
После доработки 20 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

В данной работе исследуется марковская модель киберугроз, действующих на компьютерную систему. В рамках данной модели компьютерная система рассматривается как система с отказами и восстановлениями по аналогии с моделями теории надежности. Для оценки функционально-временных свойств системы мы вводим ее параметр, называемый временем жизни и определяемый как число переходов в соответствующей марковской цепи до первого попадания в финальное состояние. В силу того, что данная случайная величина играет важную роль при оценке уровня защищенности компьютерной системы, мы подробно исследуем ее распределение вероятностей в случае несовместных киберугроз; в частности, мы получаем явные аналитические формулы для ее числовых характеристик: математического ожидания и дисперсии. Затем мы существенно обобщаем рассматриваемую марковскую модель, исключив допущение о несовместности действующих на систему киберугроз. Соответствующая марковская цепь при такой модификации расширяется за счет дополнительных состояний, не меняя своей качественной структуры. Указанный факт позволил обобщить полученные ранее аналитические результаты для математического ожидания и дисперсии времени жизни на случай совместных киберугроз. В заключении работы марковская модель совместных кибеугроз используется для постановки задачи о поиске оптимальной конфигурации средств защиты информации в заданном пространстве киберугроз. Существенно, что сформулированные оптимизационные задачи принадлежат к классу задач нелинейного дискретного (булева) программирования. В заключении работы рассматривается пример, иллюстрирующий решение задачи о выборе оптимального набора средств защиты для компьютерной системы.

Ключевые слова: киберугроза; марковская цепь; средство защиты информации; оптимизация.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Адиль Аскарович Касенов	orcid.org/0000-0002-2770-1144. E-mail: kassenov_adil@mail.ru магистрант.
Алексей Анатольевич Магазев автор для корреспонденции	orcid.org/0000-0002-8725-9183. E-mail: magazev@omgtu.ru доктор физ.-мат. наук, профессор.
Валерия Федоровна Цырульник	orcid.org/0000-0002-6875-7216. E-mail: lera.tsyrulnik@mail.ru аспирантка.

Финансирование: Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-37-90122.

Для цитирования: A. A. Kassenov, A. A. Magazev, and V. F. Tsyrulnik, “A Markov Model of Non-Mutually Exclusive Cyber Threats and its Applications for Selecting an Optimal Set of Information Security Remedies”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 108-123, 2020.

© Касенов А. А., Магазев А. А., Цырульник В. Ф., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Введение

В связи с высокой стоимостью проведения натурных экспериментов, математическое моделирование является едва ли не единственной альтернативой в исследовании проблем информационной безопасности современных компьютерных систем. Как следствие, разработка и анализ моделей компьютерной безопасности — это бурно развивающаяся область знаний, в которой число научных публикаций продолжает расти из года в год.

Из всего многообразия существующих моделей безопасности следует выделить группу *теоретико-вероятностных моделей*, основанных на различных концепциях теории вероятности и теории случайных процессов. Среди них особую роль играют модели, основанные на теории марковских случайных процессов, так как хорошо разработанный соответствующий математический аппарат во многих ситуациях позволяет получить исчерпывающее численное или даже аналитическое решение сформулированных с их помощью задач. Для иллюстрации этого тезиса достаточно отметить чрезвычайно широкой спектр приложений марковских моделей к проблемам кибербезопасности: обнаружение вторжений и аномалий в компьютерных системах [1—4], моделирование процессов распространения компьютерных вирусов [5—8], управление рисками информационной безопасности [9, 10], моделирование процессов возникновения киберугроз и эксплуатации уязвимостей в информационных и кибер-физических системах [11—13].

В работе [14] был предложен класс моделей киберугроз, формулируемых в терминах марковских цепей с дискретным временем. В рамках данных моделей компьютерная система, подвергающаяся воздействию киберугроз, описывается как система с отказами и восстановлениями (по аналогии с моделями технических систем в теории надежности). Высказав возможность использования подобных моделей для получения оценок защищенности информации, автор цитируемой работы провел лишь их поверхностный анализ и ограничился, в основном, рассмотрением простейших примеров. Частично данный недостаток был устранен в статьях [15, 16], в которых было проведено более углубленное и детальное исследование указанного класса моделей. Помимо явных аналитических формул для вероятностей состояний системы, в этих работах также был предложен оригинальный метод оценки защищенности компьютерной системы, основанный на вычислении так называемого *времени релаксации* соответствующей марковской цепи. Кроме того, с помощью исследуемой марковской модели киберугроз в работе [16] была сформулирована задача о поиске *оптимального* набора средств защиты информации, то есть набора, имеющего минимальную стоимость, но обеспечивающего необходимый уровень защиты от заданных киберугроз.

Отметим, что рассмотренные в работах [14—16] модели киберугроз сформулированы с использованием ряда упрощающих допущений, которые далеко не всегда имеют место на практике. Одним из таких допущений является предположение о том, что одновременное появление двух и более киберугроз невозможно, то есть угрозы являются *несовместными* случайными событиями. Кроме того, время релаксации марковской цепи, введенное в [15] для оценки времени достижения поглощающего состояния, представляет из себя довольно искусственную характеристику, вычисление которой осуществляется не аналитически, а численно. Цель настоящей статьи состоит в устранении этих двух недостатков. В частности, вместо времени релаксации марковской цепи мы предлагаем использовать ее более естественный параметр — *время жизни* системы, то есть число переходов в марковской цепи до достижения ее финального состояния. В настоящей работе мы подробно исследуем распределение этой случайной величины и получаем явные формулы для вычисления ее основных характеристик — математического ожидания и дисперсии. Также мы существенно обобщаем класс рассмотренных в [14, 15] марковских моделей, допустив, что все киберугрозы являются совместными случайными событиями. При этом все аналитические результаты, полученные при предположении о несовместности угроз, легко обобщаются на совместный случай с помощью расширения множества состояний марковской цепи.

В заключении настоящей статьи мы обсуждаем применение марковской модели совместных киберугроз к формулировке задачи поиска оптимальной конфигурации средств защиты информации. Данная задача имеет важное прикладное значение в управлении информационной безопасностью, в частности, в вопросах оптимизации инвестиций в кибербезопасность (см. обзорную статью [17] и приведенные в ней ссылки). В частности, мы формулируем две задачи условной оптимизации, в которых целевой функцией является либо стоимость набора средств защиты, либо среднее время жизни системы. Существенно, что обе эти задачи относятся к классу оптимизационных задач нелинейного дискретного программирования, в связи с чем актуальной становится задача поиска подходов к их эффективному решению. Разработка соответствующих методов, однако, будет представлять для нас дальнейший исследовательский интерес; здесь мы лишь ограничились рассмотрением одного простого примера, иллюстрирующего применение предложенного нами подхода к задаче выбора оптимальной конфигурации средств защиты в компьютерных системах.

1. Описание исходной модели

В настоящем разделе мы напомним основные положения модели киберугроз, предложенной в [14], а также приведем соответствующие аналитические результаты, полученные в наших предыдущих работах [15, 16].

Рассмотрим компьютерную систему (далее просто *систему*), которая подвергается воздействию n угроз с вероятностями $q_1, q_2, \dots, q_n$ соответственно. Примем следующие допущения:

- угрозы действуют на систему только в дискретные моменты времени $t = 1, 2, 3, \dots$;
- в каждый момент времени на систему может действовать только одна угроза;
- если в момент времени t на систему подействовала одна из угроз, в следующий момент $t + 1$ происходит попытка ее отражения (воздействие еще каких-либо угроз в этот момент считается невозможным).

Согласно сделанным предположениям мы можем считать, что в каждый момент времени система находится в одном из состояний $s_0, s_1, \dots, s_{n+1}$. Состояние s_0, которое мы далее будем называть *безопасным*, характеризуется отсутствием действия любой из угроз. В случае действия i-ой угрозы система переходит в состояние s_i, где $i = 1, 2, \dots, n$. Наконец, состояние s_{n+1} отвечает факту неудачного отражения любой из угроз. Данное состояние мы будем называть *финальным*.

Обозначим через r_i вероятность успешного отражения i-ой угрозы, а через $\bar{r}_i = 1 - r_i$ — вероятность соответствующей безуспешной попытки. Нетрудно видеть, что состояние системы в каждый момент времени определяется только ее состоянием в предыдущий момент времени. Это означает, что последовательность состояний системы представляет собой простую марковскую цепь, граф переходов которой изображен на рис. 1.

Задача описания динамики рассматриваемой системы сводится к вычислению величин $p_i(t)$ — вероятностей состояний s_i системы в произвольный момент времени t. Как хорошо известно из общей теории марковских цепей, эти вероятности могут быть вычислены согласно формуле

$$p_i(t) = \sum_{j=0}^{n+1} \pi_{ji}\, p_j(t-1), \qquad i = 0, 1, \dots, n+1, \tag{1}$$

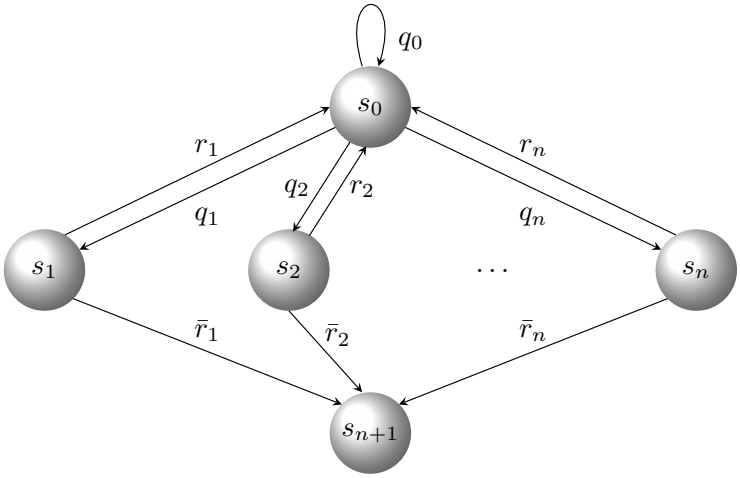


Fig. 1. System transitions graph

Рис. 1. Граф переходов системы

где π_{ji} — вероятность перехода системы из состояния s_j в состояние s_i. Совокупность величин π_{ji} образует *матрицу переходных вероятностей* Π, которая в нашем случае имеет вид:

$$\Pi = \begin{pmatrix} q_0 & q_1 & q_2 & \dots & q_n & 0 \\ r_1 & 0 & 0 & \dots & 0 & \bar{r}_1 \\ r_2 & 0 & 0 & \dots & 0 & \bar{r}_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ r_n & 0 & 0 & \dots & 0 & \bar{r}_n \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}. \tag{2}$$

Здесь введено обозначение $q_0 = 1 - \sum_{i=1}^{n} q_i$. Естественно также предположить, что в начальный момент времени $t = 0$ система находится в безопасном состоянии:

$$p_0(0) = 1, \quad p_1(0) = p_2(0) = \dots = p_{n+1}(0) = 0. \tag{3}$$

Формула (1) и начальные условия (3) позволяют однозначно определить вероятности $p_i(t)$ состояний системы в произвольный момент времени.

Выражение (1) представляет собой рекуррентную формулу, выражающую вероятность состояния s_i через вероятности состояний системы в предыдущий момент времени. Для практических целей более удобными являются явные выражения для вероятностей $p_i(t)$, рассматриваемые как функции времени t. Такие выражения были получены в нашей предыдущей работе [15]. Мы приведем здесь только вид функции $p_0(t)$, так как для дальнейших рассуждений ее будет достаточно:

$$p_0(t) = \frac{1}{w}\left(\frac{q_0 + w}{2}\right)^{t+1} - \frac{1}{w}\left(\frac{q_0 - w}{2}\right)^{t+1}. \tag{4}$$

Здесь неотрицательный параметр w определяется как

$$w^2 = q_0^2 + 4\sum_{i=1}^{n} q_i r_i. \tag{5}$$

Рассмотрим три частных случая.

1. Случай отсутствия угроз: $q_i = 0$ для всех i. Согласно (5) в этом случае $q_0 = w = 1$, поэтому в соответствии с (4) имеем

$$p_0(t) = 1.$$

Полученный результат иллюстрирует следующий тривиальный факт: при отсутствии угроз система всегда будет находиться в безопасном состоянии.

2. Случай отсутствия защиты: $r_i = 0$ для всех i. Из (5) следует, что $w = q_0$, так что формула (4) дает

$$p_0(t) = q_0^t.$$

Таким образом, вероятность безопасного состояния монотонно убывает с течением времени (здесь предполагается, что $0 < q_0 < 1$).

3. Случай частых угроз $q_0 \approx 0$. В этом случае мы приближенно можем считать $w^2 \approx 4\sum_{i=1}^{n} q_i r_i$, откуда

$$p_0(t) \approx \frac{[1 + (-1)^t]}{2} \left(\sum_{i=1}^{n} q_i r_i \right)^{t/2}.$$

Видно, что в рамках данного приближения система в нечетные моменты времени практически никогда не обнаруживается в безопасном состоянии, так как в эти моменты времени на систему с большой вероятностью воздействует какая-либо из угроз.

2. Время жизни системы: случай несовместных киберугроз

Временем жизни T системы назовем время, за которое она перейдет в финальное состояние s_{n+1}. Другими словами, время жизни — это число переходов между состояниями системы до тех пор, пока она в первый раз не окажется в состоянии s_{n+1}. Ясно, что T — это дискретная случайная величина, принимающая целые значения $T = 2, 3, 4, \ldots$. Задачей настоящего раздела является нахождение явного вида этого распределения и вычисление его основных числовых характеристик.

Закон распределения для времени жизни можно найти, используя формулу (4) для вероятности безопасного состояния $p_0(t)$. Обозначим $P(T)$ вероятность перехода системы в конечное состояние s_{n+1} ровно за T шагов. С помощью графа переходов, изображенного на рис. 2, видно, что система может оказаться в состоянии s_{n+1} за T шагов только в том случае, если в момент времени $t = T - 2$ она находилась в безопасном состоянии s_0. Так как вероятность этого события равна $p_0(T-2)$, для вероятности $P(T)$ при $T \geq 2$ имеем:

$$P(T) = p_0(T-2) \sum_{i=1}^{n} q_i \bar{r}_i.$$

Здесь выражение $\sum_{i=1}^{n} q_i \bar{r}_i$ определяет вероятность перехода из состояния s_0 в состояние s_{n+1}. С учетом (4) получаем, что распределение вероятностей случайной величины T имеет вид:

$$P(T) = \begin{cases} w^{-1} \sum\limits_{i=1}^{n} q_i \bar{r}_i \left[\left(\frac{q_0+w}{2} \right)^{T-1} - \left(\frac{q_0-w}{2} \right)^{T-1} \right], & T \geq 2, \\ 0, & T < 2. \end{cases} \tag{6}$$

Напомним, что $\bar{r}_i = 1 - r_i$, $q_0 = 1 - \sum_{i=1}^{n} q_i$, а параметр w определяется формулой (5). В качестве иллюстрации на рис. 2 приведен вид этого распределения для случая трех киберугроз.

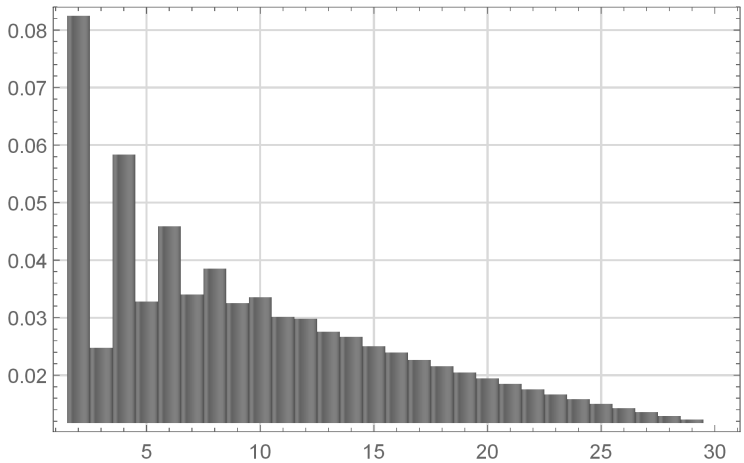


Fig. 2. Probability distribution of T for $q_1 = 0{,}35$, $q_2 = 0{,}25$, $q_3 = 0{,}1$ and $r_1 = 0{,}85$, $r_2 = 0{,}9$, $r_3 = 0{,}95$

Рис. 2. Распределение вероятностей величины T при $q_1 = 0{,}35$, $q_2 = 0{,}25$, $q_3 = 0{,}1$ и $r_1 = 0{,}85$, $r_2 = 0{,}9$, $r_3 = 0{,}95$

Напомним, что *моментом* k-го порядка случайной величины T называется математическое ожидание величины T^k:

$$\mu_k[T] = \sum_{T=0}^{\infty} T^k P(T), \quad k = 1, 2, \ldots.$$

Подставляя сюда формулу (6), получаем

$$\mu_k[T] = w^{-1} \sum_{i=1}^{n} q_i \bar{r}_i \left[\sum_{T=2}^{\infty} T^k \left(\frac{q_0 + w}{2} \right)^{T-1} - \sum_{T=2}^{\infty} T^k \left(\frac{q_0 - w}{2} \right)^{T-1} \right] =$$

$$= \frac{\sum_{i=1}^{n} q_i \bar{r}_i}{w} \left[\frac{2}{q_0 + w} \sum_{T=0}^{\infty} T^k \left(\frac{q_0 + w}{2} \right)^{T} - \frac{2}{q_0 - w} \sum_{T=0}^{\infty} T^k \left(\frac{q_0 - w}{2} \right)^{T} \right]. \quad (7)$$

В силу того, что $|q_0 \pm w| < 2$, ряды в квадратных скобках в правой части формулы (7) сходятся. Применяя известный результат (см. [18], стр. 555)

$$\sum_{n=0}^{\infty} n^k x^n = S_k(x) \equiv \left(x \frac{d}{dx} \right)^k \frac{1}{1 - x}, \quad (8)$$

мы можем записать

$$\mu_k[T] = \frac{\sum_{i=1}^{n} q_i \bar{r}_i}{w} \left[\frac{2}{q_0 + w} S_k\left(\frac{q_0 + w}{2} \right) - \frac{2}{q_0 - w} S_k\left(\frac{q_0 - w}{2} \right) \right]. \quad (9)$$

Формулы (8) и (9) позволяют выписать моменты случайной величины T для любого порядка k. В частности, момент 1-го порядка — это *математическое ожидание* $\mathbb{M}[T]$ случайной величины T. Так как $S_1(x) = x/(1 - x)^2$, из (9) получаем

$$\mathbb{M}[T] = \frac{1 + \sum_{i=1}^{n} q_i}{\sum_{i=1}^{n} q_i (1 - r_i)}. \quad (10)$$

Легко заметить, что полученная нами формула для $\mathbb{M}[T]$ вполне согласуется с ожидаемыми результатами в простейших частных случаях. Например, если $q_i = 0$ для всех i или $r_i = 1$ для всех i,

среднее время жизни становится бесконечным. Эти предельные ситуации отвечают случаю полного отсутствия угроз или случаю абсолютной защиты соответственно.

Аналогично, *дисперсия* $\mathbb{D}[T]$ случайной величины T определяется с помощью ее момента 2-го порядка следующим образом: $\mathbb{D}[T] = \mu_2[T] - \mathbb{M}[T]^2$. В силу того, что $S_2(x) = x(1+x)/(1-x)^3$, из (9) и (10) получаем:

$$\mathbb{D}[T] = \frac{1 - \sum_{i=1}^n q_i + \sum_{i=1}^n q_i r_i \left(3 + \sum_{j=1}^n q_j\right)}{\left[\sum_{i=1}^n q_i(1-r_i)\right]^2}. \tag{11}$$

Видно, что, если все q_i равны нулю или все r_i равны единице, дисперсия также как и математическое ожидание становится бесконечной.

В отсутствие защиты, то есть когда $r_i = 0$ для всех i, мы имеем

$$\mathbb{M}[T] = \frac{1}{\sum_{i=1}^n q_i} + 1, \quad \mathbb{D}[T] = \frac{1}{\sum_{i=1}^n q_i}\left(\frac{1}{\sum_{i=1}^n q_i} - 1\right).$$

Удобно выразить эти формулы через параметр $q_0 = 1 - \sum_{i=1}^n q_i$, представляющий собой вероятность отсутствия киберугроз:

$$\mathbb{M}[T] = \frac{2 - q_0}{1 - q_0}, \quad \mathbb{D}[T] = \frac{q_0}{(1-q_0)^2}.$$

Еще одна крайняя ситуация — случай частых угроз: $\sum_{i=1}^n q_i \approx 1$. Нетрудно видеть, что в этом случае

$$\mathbb{M}[T] \approx \frac{2}{1 - \sum_{i=1}^n q_i r_i}, \quad \mathbb{D}[T] \approx \frac{4\sum_{i=1}^n q_i r_i}{\left(1 - \sum_{i=1}^n q_i r_i\right)^2}.$$

3. Время жизни системы: случай совместных угроз

Ситуации, в которых на систему единовременно может воздействовать *только одна* угроза из некоторого списка возможных, представляются, на самом деле, весьма искусственными. На практике зачастую имеет место более общая картина, когда не исключаются случаи *одновременного* появления двух и более угроз, направленных на компьютерную систему. Описанная выше марковская модель киберугроз допускает естественное обобщение на указанные ситуации, приводя при этом к чисто техническим модификациям полученных в предыдущем разделе формул.

Итак, допустим теперь, что если система находится в безопасном состоянии s_0, на нее *единовременно* может воздействовать произвольный поднабор из набора n независимых киберугроз с вероятностями $q_1, q_2, \ldots, q_n$. Для описания возможных исходов удобно ввести следующую нотацию. Обозначим через x_i булеву переменную, равную 1, если в данный момент времени подействовала i-ая угроза, и равную 0 в обратном случае. Итоговый результат мы можем изобразить n-мерным булевым вектором $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, у которого единицы стоят в позициях, отвечающих номерам появившихся в данный момент времени угроз. Таким образом, система, находящаяся в момент t в состоянии s_0, в момент $t+1$ оказывается в состоянии $s_{\sigma(\mathbf{x})}$, где $\sigma(\mathbf{x}) = \sum_{i=1}^n 2^{n-i} x_i$ — десятичная форма записи булева вектора $\mathbf{x} \in \{0,1\}^n$. Считая угрозы не влияющими друг на друга, нетрудно оценить вероятность перехода из состояния s_0 в состояние $s_{\sigma(\mathbf{x})}$:

$$Q_{\sigma(\mathbf{x})} = \prod_{i=1}^n \left[x_i q_i + (1 - x_i)(1 - q_i)\right]. \tag{12}$$

Здесь i-ый сомножитель в произведении в правой части данной формулы равен q_i, если i-ая угроза подействовала в момент t, и $1 - q_i$ — в обратном случае.

Далее, если в некоторый момент t система находится в состоянии $s_{\sigma(\mathbf{x})}$, где $\mathbf{x} \neq 0$, в момент $t+1$ мы имеем два возможных исхода:

– все угрозы ликвидированы и система возвращается в безопасное состояние s_0;

– *какая-либо* из угроз успешно реализовалась и система переходит в финальное состояние s_{2^n}.

Нетрудно видеть, что вероятности $R_{\sigma(\mathbf{x})}$ и $\bar{R}_{\sigma(\mathbf{x})}$ этих двух исходов равны

$$R_{\sigma(\mathbf{x})} = \prod_{i=1}^{n} \left[x_i(r_i - 1) + 1\right], \quad \bar{R}_{\sigma(\mathbf{x})} = 1 - R_{\sigma(\mathbf{x})}, \tag{13}$$

где параметр r_i означает вероятность успешного отражения i-ой угрозы.

С учетом вышесказанного, последовательность переходов между состояниями рассматриваемой нами системы представляет собой простую марковскую цепь с матрицей переходных вероятностей вида:

$$\Pi = \begin{pmatrix} Q_0 & Q_1 & Q_2 & \dots & Q_{2^n} & 0 \\ R_1 & 0 & 0 & \dots & 0 & \bar{R}_1 \\ R_2 & 0 & 0 & \dots & 0 & \bar{R}_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ R_{2^n} & 0 & 0 & \dots & 0 & \bar{R}_{2^n} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}. \tag{14}$$

Сравнение с матрицей (2) показывает, что данная марковская цепь получается из описанной нами в предыдущем разделе марковской цепи формальной заменой:

$$n \longrightarrow 2^n, \quad q_i \longrightarrow Q_i, \quad r_i \longrightarrow R_i. \tag{15}$$

В частности, распределение случайной величины T и ее числовые характеристики могут быть получены из приведенных выше результатов подстановкой (15). В качестве примера выпишем явные аналитические выражения для среднего времени жизни $\mathbb{M}[T]$ и дисперсии $\mathbb{D}[T]$, получаемые из формул (10) и (11) с помощью замены (15):

$$\mathbb{M}[T] = \frac{1 + \sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})}}{\sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})}\left(1 - R_{\sigma(\mathbf{x})}\right)}, \tag{16}$$

$$\mathbb{D}[T] = \frac{1 - \sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})} + \sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})} R_{\sigma(\mathbf{x})} \left(3 + \sum\limits_{\mathbf{x}'\neq 0} Q_{\sigma(\mathbf{x}')}\right)}{\left[\sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})}(1 - R_{\sigma(\mathbf{x})})\right]^2}. \tag{17}$$

Здесь $Q_{\sigma(\mathbf{x})}$ и $R_{\sigma(\mathbf{x})}$ определяются формулами (12) и (13) соответственно, а суммирования осуществляются по всевозможным ненулевым векторам $\mathbf{x}$ и $\mathbf{x}'$ из $\{0,1\}^n$.

Полученные в настоящем разделе результаты были проверены с помощью численных экспериментов. Для этого с помощью пакета математических программ MatLAB была разработана имитационная модель, позволяющая получать различные реализации марковской цепи с матрицей переходных вероятностей (14). На основе статистической обработки N реализаций марковской цепи ($N \approx 100000$) мы получили численные оценки для величин $\mathbb{M}[T]$ и $\mathbb{D}[T]$ в случаях одной и двух угроз, которые затем сравнивались с теоретическими оценками, предсказываемыми формулами (16) и (17). Результаты этого сравнения для некоторых значений параметров модели приведены в таблицах 1 и 2. Из таблиц видно, что теоретические и экспериментальные результаты хорошо согласуются друг с другом.

Table 1. Expected value and variance of lifetime T for one cyber threat

Таблица 1. Математическое ожидание и дисперсия времени жизни T в случае одной киберугрозы

Параметры модели		**Математическое ожидание** $\mathbb{M}[T]$		**Дисперсия** $\mathbb{D}[T]$	
q	r	**Теория**	**Эксперимент**	**Теория**	**Эксперимент**
0,2000	0,8000	30,0000	30,0235	820,0000	819,7795
0,4000	0,5000	7,0000	7,0005	32,0000	31,6664
0,7000	0,3000	3,4693	3,4710	4,48563	4,6000

Table 2. Expected value and variance of lifetime T for two cyber threats

Таблица 2. Математическое ожидание и дисперсия времени жизни T в случае двух киберугроз

Параметры модели				**Математическое ожидание** $\mathbb{M}[T]$		**Дисперсия** $\mathbb{D}[T]$	
q_1	q_2	r_1	r_2	**Теория**	**Эксперимент**	**Теория**	**Эксперимент**
0,2000	0,5000	0,8000	0,7000	8,6956	8,7331	56,0491	55,9479
0,4000	0,2000	0,5000	0,5000	5,4285	5,4352	16,8980	16,8045
0,7000	0,2000	0,3000	0,7000	3,3807	3,3765	4,2068	4,1692

4. Оптимизация выбора средств защиты информации

В работе [15] была высказана идея об использовании описанной в разделе 1 марковской модели киберугроз в задаче о выборе оптимального набора средств защиты информации. Более подробно эта идея обсуждается в [16]. В данной главе мы напомним основную постановку соответствующей оптимизационной задачи применительно к модифицированной версии модели с совместными киберугрозами.

Допустим, что для отражения существующих киберугроз имеется набор m различных средств защиты. Обозначим через z_a булеву переменную, ассоциированную с a-ым средством защиты: $z_a = 1$, если a-ое средство используется, и $z_a = 0$ — в обратном случае. Таким образом, мы имеем множество из 2^m возможных конфигураций системы защиты информации; каждая конфигурация будет описываться булевым вектором $\mathbf{z} = (z_1, z_2, \dots, z_m) \in \{0,1\}^m$. В частности, нулевому $\mathbf{z}$ будет отвечать конфигурация, в которой никакие средства не задействованы, а случай $\mathbf{z} = (1,1,\dots,1)$ отвечает использованию всех имеющихся средств защиты информации.

Обозначим через $r_{i,a}$ вероятность отражения i-ой угрозы a-ым средством защиты. В общем случае одну и ту же угрозу могут отражать сразу несколько средств защиты, поэтому вероятность отражения i-ой угрозы *хотя бы одним* средством защиты определяется в соответствии с формулой (см. [19], стр. 99):

$$r_i(\mathbf{z}) = \sum_{k=1}^{m} (-1)^k \sum_{a_1<a_2<\dots<a_k} (r_{i,a_1} z_{a_1})(r_{i,a_2} z_{a_2}) \dots \left(r_{i,a_k} z_{a_k}\right). \tag{18}$$

Подстановка этих выражений в формулу (13) позволяет получить вероятности отражения действующих поднаборов киберугроз, изображаемых, следуя нотации предыдущего раздела, n-мерными булевыми векторами $\mathbf{x} = (x_1, x_2, \dots, x_n)$:

$$R_{\sigma(\mathbf{x})}(\mathbf{z}) = \prod_{i=1}^{n} \left[(1 - x_i) r_i(\mathbf{z}) + 1\right]. \tag{19}$$

Напомним, что здесь $x_i = 1$, если в данный момент времени подействовала i-ая угроза, и $x_i = 0$ — в обратном случае. Отсюда для среднего времени жизни системы мы получаем выражение, зависящее от $\mathbf{z}$:

$$\mathbb{M}[T](\mathbf{z}) = \frac{1 + \sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})}}{\sum\limits_{\mathbf{x}\neq 0} Q_{\sigma(\mathbf{x})} \left(1 - R_{\sigma(\mathbf{x})}(\mathbf{z})\right)}. \tag{20}$$

На практике довольно часто ставится задача определения *оптимального поднабора* из некоторого заранее заданного набора средств защиты информации. В зависимости от конкретных целей соответствующая задача оптимизации может быть сформулирована по-разному (см., например, [20]). Оказывается, что с использованием рассматриваемой модели мы можем сформулировать несколько задач оптимизации, сводящихся к нахождению определенного баланса между экономической стоимостью защитных мер и их функциональной эффективностью.

Обозначим через c_a стоимость a-го средства защиты (в условных денежных единицах). Тогда функция стоимости данной конфигурации системы защиты информации имеет следующий вид:

$$C(\mathbf{z}) = \sum_{a=1}^{m} c_a z_a.$$

Первая из оптимизационных задач, которую мы можем сформулировать с использованием имеющихся конструкций, звучит так: при существующих ограничениях на используещиеся при построении системы защиты ресурсы требуется максимизировать среднее время жизни компьютерной системы. Формальная запись данной оптимизационной задачи имеет следующий вид:

$$\mathbb{M}[T](\mathbf{z}) \longrightarrow \max, \quad C(\mathbf{z}) \leq C_0. \tag{21}$$

Здесь C_0 — положительная постоянная, означающая максимальную величину затрат на защиту от угроз. Вторая оптимизационная задача заключается в поиске такой конфигурации системы защиты, при которой вложения в защиту будут минимальны при имеющемся ограничении на продолжительность функционирования компьютерной системы:

$$\mathbb{M}[T](\mathbf{z}) \geq T_0, \quad C(\mathbf{z}) \longrightarrow \min. \tag{22}$$

Отметим, что обе эти задачи представляют интерес и часто встречаются при решении реальных задач при проектировании и разработке систем обеспечения информационной безопасности.

Как следует из формул (18)–(20), величина $\mathbb{M}[T](\mathbf{z})$ имеет вид $1/P(\mathbf{z})$, где $P(\mathbf{z})$ представляет собой полином степени m от булевых переменных $z_1, \dots, z_m$. Следовательно, оптимизационные задачи (21) и (22) принадлежат к классу задач нелинейного целочисленного программирования. Как известно, универсальных и эффективных алгоритмов решения подобных задач на сегодняшний день не существует. С другой стороны, как показали численные эксперименты, при небольших значениях m ($m \lesssim 15$) задачи (21) и (22) могут быть решены методом прямого перебора. При больших m определенную эффективность демонстрирует метод последовательного анализа вариантов [21], учитывающий имеющуюся специфику функций $C(\mathbf{z})$ и $\mathbb{M}[T](\mathbf{z})$. Строгая оценка вычислительной

сложности этого подхода будет представлять для нас дальнейший исследовательский интерес, а в настоящей статье мы ограничимся сделанными замечаниями и просто продемонстрируем применение изложенных нами идей на гипотетическом примере.

Пример. Рассмотрим абстрактную компьютерную систему (это может быть, например, отдельный компьютер с установленным системным и прикладным ПО или совокупность подобных компьютеров, объединенных в локальную сеть) и продемонстрируем как на основе рассмотренной выше марковской модели безопасности может быть определен оптимальный набор средств ее защиты.

При выборе наиболее актуальных угроз для данной системы мы можем воспользоваться банком данных угроз безопасности информации ФСТЭК России[1]. Ограничиваясь только угрозами, устраняемыми программными средствами защиты и характерными только для нарушителей с низким потенциалом, мы примем в качестве наиболее актуальных восемь угроз, перечисленных в таблице 3 (конечно, в реальных ситуациях их больше). В этой же таблице приведены вероятности возникновения этих угроз за единичный интервал времени $\Delta t = 1$. Отметим, что значения q_i в рассматриваемом примере носят достаточно декларативный характер; для конкретных объектов эти величины на практике получаются экспертным методом с учетом применяемых на объекте информационных технологий и программно-аппаратных средств.

Table 3. Actual threats for the described computer system and probability of their appearances for a single time interval

Таблица 3. Актуальные угрозы для рассматриваемой компьютерной системы и вероятности их появления за единичный интервал времени

№	ID	Описание угрозы	Вероятность
1	УБИ.006	Угроза внедрения кода или данных	0,02
2	УБИ.018	Угроза загрузки нештатной операционной системы	0,01
3	УБИ.031	Угроза использования механизмов авторизации для повышения привилегий	0,03
4	УБИ.034	Угроза использования слабостей протоколов сетевого/локального обмена данными	0, 03
5	УБИ.116	Угроза перехвата данных, передаваемых по вычислительной сети	0,02
6	УБИ.130	Угроза подмены содержимого сетевых ресурсов	0,04
7	УБИ.167	Угроза заражения компьютеров при посещении неблагонадежных сайтов	0,05
8	УБИ.170	Угроза неправомерного шифрования информации	0,02

В таблице 4 приведен перечень представителей классов типовых средств защиты информации, наиболее часто используемых для отражения киберугроз из таблицы 3, а также ориентировочные затраты, связанные с их приобретением и эксплуатацией[2].

В соответствии с таблицей 4 функция стоимости $C(\mathbf{z})$, определенная на множестве конфигураций системы защиты, для нашего примера имеет вид:

$$C(\mathbf{z}) = 20000z_1 + 10000z_2 + 8000z_3 + 15000z_4 + 10000z_5 + 5000z_6.$$

[1] https://bdu.fstec.ru/threat

[2] Стоимости приведены достаточно условно ввиду большого разнообразия имеющихся на современном рынке конкретных представителей различных классов средств защиты. Кроме того, в реальных системах стоимости очень сильно зависят от масштабов самой системы (числа рабочих станций, числа пользователей и т.д.), а также от срока их эксплуатации и пр.

Вероятности $r_{i,a}$ отражения угроз средствами защиты на практике обычно определяются с помощью экспертных оценок [22]. В нашем случае мы введем эти вероятности также довольно декларативно, так как в реальных ситуациях необходимо учитывать множество особенностей конкретной компьютерной системы и конкретных используемых средств защиты информации:

$$\|r_{i,a}\| = \begin{pmatrix} 0{,}8 & 0{,}5 & 0{,}25 & 0{,}5 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0{,}9 & 0 \\ 0 & 0 & 0 & 0,8 & 0{,}2 & 0 \\ 0 & 0{,}7 & 0 & 0 & 0 & 0{,}5 \\ 0 & 0{,}8 & 0 & 0 & 0 & 0{,}5 \\ 0 & 0{,}8 & 0 & 0 & 0 & 0{,}5 \\ 0{,}9 & 0{,}5 & 0 & 0 & 0 & 0 \\ 0{,}2 & 0 & 0{,}2 & 0{,}5 & 0{,}1 & 0 \end{pmatrix}. \tag{23}$$

Отметим, что в нашем примере каждая из угроз может быть отражена несколькими средствами защиты с различной степенью эффективности. Согласно данной матрице, например, угроза УБИ.167 (угроза заражения компьютеров при посещении неблагонадежных сайтов) с вероятностью 0,9 отражается имеющимся антивирусным ПО и с вероятностью 0,5 отражается межсетевым экраном с помощью различных механизмов фильтрации и анализа сетевого трафика.

Ограничимся решением оптимизационной задачи (22). Подставляя элементы матрицы (23) в формулу (18), получаем набор из $n = 8$ полиномов $r_i(\mathbf{z})$ от 6 булевых переменных $z_1, \ldots, z_6$, ассоциированных с соответствующими средствами защиты из таблицы 3:

$$\begin{aligned}
r_1(\mathbf{z}) &= 0{,}8z_1 + 0{,}5z_2 + 0{,}25z_3 + 0{,}5z_4 - 0{,}4z_1z_2 - 0{,}2z_1z_3 - 0{,}4z_1z_4 - 0{,}125z_2z_3 - \\
&\quad - 0{,}25z_2z_4 - 0{,}125z_3z_4 + 0{,}1z_1z_2z_3 + 0{,}2z_1z_2z_4 + 0{,}1z_1z_3z_4 + 0{,}0625z_2z_3z_4 - \\
&\quad - 0{,}05z_1z_2z_3z_4, \\
r_2(\mathbf{z}) &= 0{,}5z_3 + 0{,}9z_5 - 0{,}45z_3z_5, \\
r_3(\mathbf{z}) &= 0{,}8z_4 + 0{,}2z_5 - 0{,}16z_4z_5, \\
r_4(\mathbf{z}) &= 0{,}7z_2 + 0{,}5z_6 - 0{,}35z_2z_6, \\
r_5(\mathbf{z}) &= 0{,}8z_2 + 0{,}5z_6 - 0{,}4z_2z_6, \\
r_6(\mathbf{z}) &= 0{,}8z_2 + 0{,}5z_6 - 0{,}4z_2z_6, \\
r_7(\mathbf{z}) &= 0{,}9z_1 + 0{,}5z_2 - 0{,}45z_1z_2, \\
r_8(\mathbf{z}) &= 0{,}2z_1 + 0{,}2z_3 + 0{,}5z_4 + 0{,}1z_5 - 0{,}04z_1z_3 - 0{,}1z_1z_4 - 0{,}02z_1z_5 - 0{,}02z_3z_5 - \\
&\quad - 0{,}1z_3z_4 - 0{,}05z_4z_5 + 0{,}02z_1z_3z_4 + 0{,}004z_1z_3z_5 + 0{,}01z_1z_4z_5 + 0{,}01z_3z_4z_5 - \\
&\quad - 0{,}002z_1z_3z_4z_5.
\end{aligned}$$

После подстановки полиномов $r_i(\mathbf{z})$ в формулу (19), находим 2^n полиномов $R_{\sigma(\mathbf{x})}(\mathbf{z})$[3], определяющих вероятности отражения поднаборов $\mathbf{x}$ киберугроз данной конфигурацией средств защиты $\mathbf{z}$. Далее, используя вероятности возникновения угроз q_i, приведенные в таблице 3, получаем в соответствии с формулой (12):

$$Q_{\sigma(\mathbf{x})} = (x_1 - 0{,}98)(x_2 - 0{,}99)(x_3 - 0{,}97)(x_4 - 0{,}97)(x_5 - 0{,}98) \times (x_6 - 0{,}96)(x_7 - 0{,}95)(x_8 - 0{,}98).$$

[3]Напомним, что $\sigma(\mathbf{x}) = \sum_{i=1}^{n} 2^{n-i}x_i$.

Table 4. Security remedies against current cyber threats and their costs

Таблица 4. Средства защиты от актуальных киберугроз и их стоимости

№	Средство защиты	Стоимость c_a в усл. ед.
1	Средство антивирусной защиты	20 000
2	Программный межсетевой экран	10 000
3	Средство защиты от НСД	8 000
4	Система разграничения доступа	15 000
5	Средство доверенной загрузки	10 000
6	Средство криптографической защиты информации	5 000

Подставляя полученные выражения для $R_{\sigma(\mathbf{x})}(\mathbf{z})$ и $Q_{\sigma(\mathbf{x})}$ в формулу (16), после суммирования по всевозможным $\mathbf{x} \in \{0,1\}^n$ находим среднее время жизни нашей системы в виде $\mathbb{M}[T](\mathbf{z}) = P^{-1}(\mathbf{z})$, где $P(\mathbf{z})$ — полином шестой степени от булевых переменных $z_1, \ldots, z_6$ (мы не выписываем его здесь в виду громоздкости). На рис. 3 приведена графическая зависимость среднего времени жизни системы от номера конфигурации $\sigma(\mathbf{z})$. Из рисунка видно, что минимальное среднее время жизни системы соответствует нулевой конфигурации $\sigma(\mathbf{z}) = 0$ ($\mathbb{M}[T]_{\min} = 5{,}9891$), а максимальное — конфигурации $\sigma(\mathbf{z}) = 2^m - 1 = 64$ ($\mathbb{M}[T]_{\max} = 48{,}8869$).

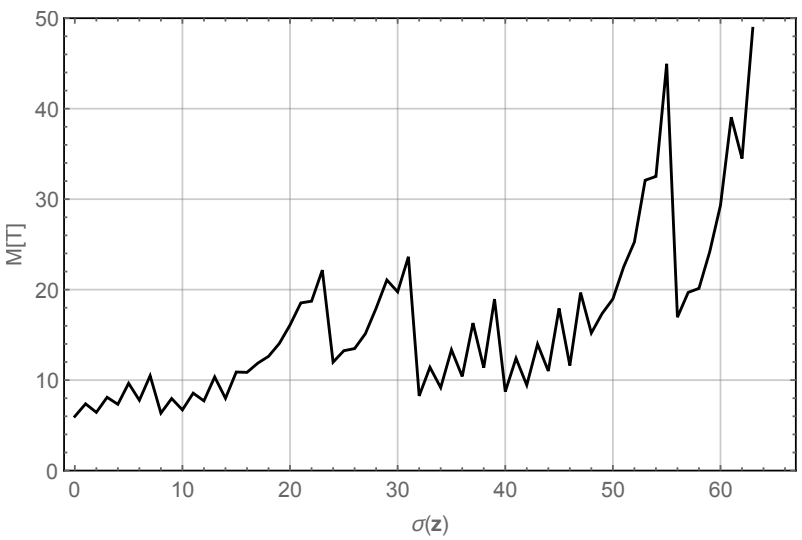


Fig. 3. Dependence of $\mathbb{M}[T]$ on $\sigma(\mathbf{z})$ for the considered example

Рис. 3. Зависимость $\mathbb{M}[T]$ от $\sigma(\mathbf{z})$ для рассматриваемого примера

Перейдем к решению оптимизационной задачи (22). Так как число возможных вариантов здесь невелико и равно $2^m = 64$, мы решали эту задачу методом прямого перебора. В таблице 5 для каждого рассматриваемого значения T_0 приводится найденное оптимальное решение $\mathbf{z}$ и соответствующая стоимость $C(\mathbf{z})$. Как видно из таблицы, стоимость оптимальной конфигурации средств защиты увеличивается с ростом T_0, что, очевидно, согласуется с реальным положением вещей, так как более длительное безотказное функционирование системы требует больших затрат.

В заключение отметим, что цель данного примера — демонстрация использования рассмотренной выше марковской модели киберугроз для формулировки задачи о выборе оптимального поднабора средств защиты информации. Данный пример никоим образом не исчерпывает всего

Table 5. Solutions of the optimization task (22) for the different values of T_0

Таблица 5. Решения оптимизационной задачи (22) для различных значений T_0

T_0	**Оптимальная конфигурация**	**Стоимость**, руб.
10	$\mathbf{z} = (0, 1, 0, 0, 0, 0)$	10 000
15	$\mathbf{z} = (0, 1, 0, 1, 0, 0)$	25 000
20	$\mathbf{z} = (0, 1, 1, 1, 0, 1)$	38 000
25	$\mathbf{z} = (1, 1, 0, 1, 0, 0)$	45 000
30	$\mathbf{z} = (1, 1, 0, 1, 0, 1)$	50 000
35	$\mathbf{z} = (1, 1, 1, 1, 0, 1)$	58 000
40	$\mathbf{z} = (1, 1, 0, 1, 1, 1)$	60 000

многообразия возникающих на практике ситуаций с системами защиты информации и служит скорее иллюстрацией одного из возможных подходов к оценке уровня защищенности информации в современных информационных системах.

Заключение

В настоящей статье мы продолжили исследование класса марковских моделей киберугроз, начатое в предыдущих работах [14—16]. В рамках этих моделей компьютерная система, подвергающаяся действию киберугроз, рассматривается как система с отказами и восстановлениями, функционируя до момента своего полного (фатального) отказа. В настоящей работе мы ввели понятие времени жизни системы, определяя его как число переходов в соответствующей марковской цепи до первого попадания в финальное состояние. Получив явную формулу для распределения данной случайной величины, мы также вычислили ее явные числовые характеристики — математическое ожидание и дисперсию. Далее мы существенно обобщили рассматриваемый класс марковских моделей, отказавшись от допущения о несовместности киберугроз. В заключение с помощью марковской модели совместных киберугроз мы сформулировали две задачи нелинейного дискретного программирования о нахождении оптимального набора средств защиты. В качестве примера рассмотрена задача о выборе оптимальной конфигурации средств защиты для простейшей компьютерной системы с восемью актуальными кибеугрозами.

Наши дальнейшие исследовательские перспективы будут связаны с ослаблением допущения о независимости киберугроз, а также с разработкой эффективных подходов к решению сформулированных в разделе 4 оптимизационных задач. Отметим также, что полученные нами результаты могут быть использованы в различных методиках и стандартах, посвященных оценке и анализу защищенности современных компьютерных систем и вычислительных сетей.

References

[1] N. Ye, Y. Zhang, and B. C.M., “Robustness of the Markov-chain model for cyber-attack detection”, *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 116–123, 2004.

[2] S. Jha, K. Tan, and R. Maxion, “Markov Chains, Classifiers, and Intrusion Detection.”, in *Proc. IEEE Computer Security Foundations Workshops*, vol. 1, 2001, pp. 206–219.

[3] A. Ahmadian Ramaki, A. Rasoolzadegan, and A. Javan Jafari, “A systematic review on intrusion detection based on the Hidden Markov Model”, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 11, no. 3, pp. 111–134, 2018.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges”, *Computers and Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[5] L. Billings, W. Spears, and I. Schwartz, "A unified prediction of computer virus spread in connected networks", *Physics Letters A*, vol. 297, no. 3-4, pp. 261–266, 2002.

[6] A. Boyko, "Sposob analiticheskogo modelirovaniya protsessa rasprostraneniya virusov v komp'yuternykh setyakh razlichnoy struktury", *Trudy SPIIRAN*, vol. 5, no. 42, pp. 196–211, 2015.

[7] Y. Dalinger, D. Babanin, and B. S.M., "Matematicheskie modeli rasprostraneniya virusov v komp'yuternykh setyakh razlichnoy struktury", *Informatika i sistemy upravleniya*, no. 4, pp. 25–33, 2012.

[8] A. Del Rey, "Mathematical modeling of the propagation of malware: a review", *Security and Communication Networks*, vol. 8, no. 15, pp. 2561–2579, 2015.

[9] M. Yang, R. Jiang, T. Gao, W. Xie, and J. Wang, "Research on Cloud Computing Security Risk Assessment Based on Information Entropy and Markov Chain.", *I. J. Network Security*, vol. 20, no. 4, pp. 664–673, 2018.

[10] C. Xiaolin, T. Xiaobin, Z. Yong, and X. Hongsheng, "A Markov game theory-based risk assessment model for network information system", in *International Conference on Computer Science and Software Engineering, China*, IEEE, vol. 3, 2008, pp. 1057–1061.

[11] H. Orojloo and M. Azgomi, "A method for modeling and evaluation of the security of cyber-physical systems", in *11th International ISC Conference on Information Security and Cryptology, Iran*, IEEE, 2014, pp. 131–136.

[12] J. Almasizadeh and M. Azgomi, "A stochastic model of attack process for the evaluation of security metrics", *Computer Networks*, vol. 57, no. 10, pp. 2159–2180, 2013.

[13] K. Shcheglov and A. Shcheglov, "Markovskie modeli ugrozy bezopasnosti informatsionnoy sistemy", *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie*, vol. 58, no. 12, pp. 957–965, 2015.

[14] A. Rosenko, "Matematicheskoe modelirovanie vliyaniya vnutrennikh ugroz na bezopasnost' konfidentsial'noy informatsii, tsirkuliruyushchey v avtomatizirovannoy informatsionnoy sisteme", *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki*, vol. 85, no. 8, pp. 71–81, 2008.

[15] A. Magazev and V. Tsyrulnik, "Investigation of a Markov Model for Computer System Security Threats", *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 615–624, 2018.

[16] A. Magazev and V. Tsyrulnik, "Optimizing the selection of information security remedies in terms of a Markov security model", in *Journal of Physics: Conference Series*, vol. 1096, 2018, p. 012 160.

[17] D. Shirtz and Y. Elovici, "Optimizing investment decisions in selecting information security remedies", *Information Management and Computer Security*, vol. 19, no. 2, pp. 95–112, 2011.

[18] P. A.P., Y. Brychkov, and M. O.I., *Integrals and series: Elementary functions*. Gordon&Breach Sci. Publ., New York, 1986, vol. 1.

[19] W. Feller, *An introduction to probability theory and its applications*. John Wiley & Sons Inc, 1968, vol. 1, 528 pp.

[20] A. e. a. Ovchinnikov, "Matematicheskaya model' optimal'nogo vybora sredstv zashchity ot ugroz bezopasnosti vychislitel'noy seti predpriyatiya", *Vestnik Moskovskogo gosudarstvennogo tekhnicheskogo universiteta im. N. E. Baumana. Ser. "Priborostroenie"*, no. 3, pp. 115–121, 2007.

[21] M. Kovalev, *Diskretnaya optimizatsiya (tselochislennoe programmirovanie)*, 2-e izd., stereotipnoe. Editorial URSS, 2003, 192 pp.

[22] Beshelev, S.D., and F. Gurvich, *Matematiko-statisticheskie metody ekspertnykh otsenok*. 1980, 263 pp.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 27, NO. 1, 2020
journal homepage: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

Calculation of Derivatives in the L_p Spaces where $1 \le p \le \infty$

A. N. Morozov[1]

DOI: 10.18255/1818-1015-2020-1-124-131

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 41A35, 41A45, 65D25
Research article
Full text in Russian

Received February 9, 2020
After revision February 26, 2020
Accepted February 28, 2020

It is well known in functional analysis that construction of k-order derivative in Sobolev space W_p^k can be performed by spreading the k-multiple differentiation operator from the space C^k. At the same time there is a definition of (k,p)-differentiability of a function at an individual point based on the corresponding order of infinitesimal difference between the function and the approximating algebraic polynomial k-th degree in the neighborhood of this point on the norm of the space L_p. The purpose of this article is to study the consistency of the operator and local derivative constructions and their direct calculation. The function $f \in L_p[I]$, $p > 0$, (for $p = \infty$, we consider measurable functions bounded on the segment I) is called $(k;p)$-differentiable at a point $x \in I$ if there exists an algebraic polynomial of π of degree no more than k for which holds $\|f - \pi\|_{L_p[J_h]} = o(h^{k+\frac{1}{p}})$, where $J_h = [x_0 - h; x_0 + h] \cap I$. At an internal point for $k = 1$ and $p = \infty$ this is equivalent to the usual definition of the function differentiability. The discussed concept was investigated and applied in the works of S. N. Bernshtein [1], A. P. Calderon and A. Sigmund [2]. The author's article [3] shows that uniform (k,p)-differentiability of a function on the segment I for some $p \ge 1$ is equivalent to belonging the function to the space $C^k[I]$ (existence of an equivalent function in $C^k[I]$). In present article, integral-difference expressions are constructed for calculating generalized local derivatives of natural order in the space L_1 (hence, in the spaces L_p, $1 \le p \le \infty$), and on their basis - sequences of piecewise constant functions subordinate to uniform partitions of the segment I. It is shown that for the function f from the space W_p^k the sequence piecewise constant functions defined by integral-difference k-th order expressions converges to $f^{(k)}$ on the norm of the space $L_p[I]$. The constructions are algorithmic in nature and can be applied in numerical computer research of various differential models.

Keywords: Differentiability of Function in the Spaces L_p; Differences for the Space L_1; Numerical Finding of Derivatives on a Computer; The Spreading of the Differentiation Operator

INFORMATION ABOUT THE AUTHORS

Anatoly Nikolaevich Morozov | orcid.org/0000-0001-9940-159X. E-mail: moroz@uniyar.ac.ru
PhD.

For citation: A. N. Morozov, "Calculation of Derivatives in the L_p Spaces where $1 \le p \le \infty$", *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 124-131, 2020.

© Morozov A. N., 2020
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 27, № 1, 2020
сайт журнала: www.mais-journal.ru

DISCRETE MATHEMATICS IN RELATION TO COMPUTER SCIENCE

Вычисление производных в пространствах L_p, $1 \le p \le \infty$

А. Н. Морозов[1]

DOI: 10.18255/1818-1015-2020-1-124-131

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003 Россия.

УДК 519.65
Научная статья
Полный текст на русском языке

Получена 9 февраля 2020 г.
После доработки 26 февраля 2020 г.
Принята к публикации 28 февраля 2020 г.

В функциональном анализе хорошо известно рассуждение о построении производных k-го порядка в пространствах Соболева W_p^k при помощи распространения оператора k-кратного дифференцирования с пространства C^k. В то же время имеется определение (k,p)-дифференцируемости функции в индивидуальной точке, основанное на соответствующего порядка бесконечно малом отличии функции от приближающего её алгебраического многочлена k-ой степени в окрестности этой точки по норме пространства L_p. Целью данной статьи является исследование согласованности операторного и локального построений производной и непосредственное их вычисление. Функция $f \in L_p[I]$, $p > 0$, (при $p = \infty$ рассматриваются измеримые ограниченные на отрезке I функции) называется (k,p)-дифференцируемой в точке $x \in I$, если существует алгебраический многочлен π степени не больше k, для которого выполняется $\|f - \pi\|_{L_p[J_h]} = o(h^{k+\frac{1}{p}})$, где $J_h = [x-h; x+h] \cap I$. Во внутренней точке при $k = 1$ и $p = \infty$ это равносильно определению обычной дифференцируемости функции. Обсуждаемое понятие исследовалось и применялось в работах С. Н. Бернштейна [1], А. П. Кальдерона и А. Зигмунда [2]. В статье автора [3] показано, что равномерная (k,p)-дифференцируемость функции на отрезке I при некотором $p \ge 1$, равносильна принадлежности этой функции пространству $C^k[I]$ (существованию эквивалентной функции в $C^k[I]$). В настоящей статье построены интегрально-разностные выражения для вычисления обобщённых локальных производных натурального порядка в пространстве L_1 (следовательно, в пространствах L_p, $1 \le p \le \infty$), а на их основе – последовательности кусочно-постоянных функций, подчинённых равномерным разбиениям отрезка. Показано, что для функции f из пространства W_p^k последовательность кусочно-постоянных функций, определённых посредством интегрально-разностных выражений k-го порядка, сходится к $f^{(k)}$ по норме пространства $L_p[I]$. Построения имеют алгоритмический характер, и могут быть применены в численном исследовании на ЭВМ различных дифференциальных моделей.

Ключевые слова: Дифференцируемость функции в пространствах L_p; разностные выражения для пространства L_1; численное нахождение производных на ЭВМ; распространение оператора дифференцирования

ИНФОРМАЦИЯ ОБ АВТОРАХ

Анатолий Николаевич Морозов | orcid.org/0000-0001-9940-159X. E-mail: moroz@uniyar.ac.ru
канд. физ.-мат. наук, доцент.

Для цитирования: A. N. Morozov, “Calculation of Derivatives in the L_p Spaces where $1 < p \le \infty$”, *Modeling and analysis of information systems*, vol. 27, no. 1, pp. 124-131, 2020.

© Морозов А. Н., 2020
Эта статья открытого доступа под лицензией CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Введение и основные обозначения

Как обычно, $L_p[I]$ обозначает пространство действительных измеримых функций, интегрируемых в степени p ($0 < p < \infty$) по Лебегу на отрезке $I = [a; b]$,

$$\|f\|_{L_p[I]} = \left(\int\limits_I |f(x)|^p dx \right)^{\frac{1}{p}};$$

при $p = \infty$ всюду ниже рассматривается $B[I]$ – пространство измеримых ограниченных на отрезке I функций, –

$$\|f\|_{B[I]} = \sup_{x \in I} |f(x)|.$$

Введём для краткости записи семейство пространств

$$X_p[I] = \begin{cases} L_p[I] & \text{при } p < \infty, \\ B[I] & \text{при } p = \infty. \end{cases}$$

Когда неясность исключена, сокращаем обозначения до X_p и $\|f\|_p$. Длину I обозначаем $|I|$.

Также используются ($k \in \mathbf{N}$) $C^k = C^k[I]$ – пространство k раз непрерывно дифференцируемых на отрезке I функций, – и ($1 \leq p < \infty$)

$W_p^k = W_p^k[I] = \left\{ f : f^{(k-1)} \text{ абсолютно непрерывна на отрезке } I, f^{(k)} \in L_p[I] \right\}$,

с нормами $\|f\|_p + \|f^{(k)}\|_p$.

Определение 1. *Функция $f \in X_p[I]$ называется (k, p)-дифференцируемой в точке $x \in I$, если существует алгебраический многочлен π степени не больше k, для которого выполняется*

$\|f - \pi\|_{X_p[J_{x,h}]} = o(h^{k+\frac{1}{p}})$, *при* $h \longrightarrow 0$, *где* $J_{x,h} = [x - h; x + h] \cap I$.

Такой многочлен может быть только один, его часто называют тейлоровским. Во внутренней точке x при $k = 1$ и $p = \infty$ данное определение совпадает с определением обычной дифференцируемости (существованием производной), но в общем случае из (k, ∞)-дифференцируемости в точке не следует существование k-й производной.

Классическим примером является функция

$$f(x) = \begin{cases} x^{k+1} \sin(\frac{1}{x^k}), & x \neq 0, \\ 0, & x = 0; \end{cases}$$

которая (m, ∞)-дифференцируема в нуле для $m = 1, \ldots, k$, но имеет в этой точке лишь одну обычную производную.

Следующий пример иллюстрирует особенности (k, p)-дифференцируемости при $p < \infty$. Рассмотрим для определённости чётную функцию f, $f(0) = 1$, задаваемую на $(0; 1]$ формулой

$$f(x) = \sum_{j=1}^{\infty} f_j(x),$$

где

$$f_j(x) = \begin{cases} 1, & x \in H_j = \left(\frac{1}{2^j} - \frac{1}{2^{j^2+2}}; \frac{1}{2^j} \right), \\ 0, & x \notin H_j. \end{cases}$$

Для каждого $0 < h \leq \frac{1}{2}$ пусть $n \in \mathbf{N}$ таково, что $\frac{1}{2^{n+1}} < h \leq \frac{1}{2^n}$. Если $\pi \equiv 0$, при всех $0 < p < \infty$ выполняется

$$\|f - \pi\|^p_{L_p[-h;h]} \leq \sum_{j=n}^{\infty} \frac{1}{2^{j^2+1}} < \frac{1}{2^{n^2}}.$$

Из чего следует, что f является (k, p)-дифференцируемой в нуле при любых рассматриваемых k и $p < \infty$.

С целью полноты освещения вопроса приведём следующее утверждение.

Предложение 1. *Если функция f является (k,p)-дифференцируемой $(0 < p \le \infty)$ в точке $x \in I$, то она является (m,p)-дифференцируемой в этой точке, $m = 1, \ldots, k-1$. При этом тейлоровский многочлен из условия (m,p)-дифференцируемости представляет собой соответствующую часть тейлоровского многочлена π из условия (k,p)-дифференцируемости, записанного в виде*

$$\pi(t) = a_0 + a_1(t-x) + \cdots + a_m(t-x)^m + \cdots + a_k(t-x)^k.$$

Доказательство. Без потери общности будем считать, что $x = 0$ и является левым краем отрезка I. По условию существует алгебраический многочлен $\pi(t) = a_0 + a_1 t + \cdots + a_m t^m + \cdots + a_k t^k$ такой, что $\|f - \pi\|_{L_p[0;h]} = o(h^{k+\frac{1}{p}})$ при $h \longrightarrow 0$. Пусть $p_* = \min\{p\,;1\}$. Получаем

$$\|f - (a_0 + \cdots + a_m t^m)\|^{p_*}_{L_p[0;h]} \le \|f - \pi\|^{p_*}_{L_p[0;h]} + \|a_{m+1}t^{m+1} + \cdots + a_k t^k\|^{p_*}_{L_p[0;h]}$$

$$\le \|f - \pi\|^{p_*}_{L_p[0;h]} + \sum_{j=m+1}^{k} \left(|a_j| h^{j+\frac{1}{p}}\right)^{p_*} = o\left(\left(h^{m+\frac{1}{p}}\right)^{p_*}\right).$$

□

Для (k,p)-дифференцируемой в точке x функции f будем применять обозначение $f_p^{(k)}(x) \stackrel{\text{def}}{=} k! \cdot a_k$, где a_k – коэффициент при степени k многочлена из условия (k,p)-дифференцируемости.

В работах С. Н. Бернштейна [1], А. П. Кальдерона и А. Зигмунда [2] были даны приложения такого понятия к построению описания функциональных пространств ($p = \infty$) и изучению локальных свойств решений дифференциальных уравнений ($1 \le p \le \infty$) соответственно.

В статье [3] автором рассмотрена ситуация с равномерной (k,p)-дифференцируемостью функции на отрезке. (В таком случае, в частности, определена функция $f_p^{(k)}$.)

Замечание. В этой статье доказательства проводились на основе методов локальных приближений функций алгебраическими многочленами, поэтому для выстраивания общей схемы рассуждения при $p = \infty$ в качестве базового рассматривалось пространство непрерывных на отрезке I функций $C[I]$. Из Предложения 1, приведённого выше, сразу следует (рассмотрев $m = 0$), что применительно к исследованию равномерной (k,∞)-дифференцируемости это равносильно использованию пространства $B[I]$.

Определение 2. *(k,p)-дифференцируемая $(0 < p \le \infty)$ во всех точках отрезка I функция f называется равномерно (k,p)-дифференцируемой на I, если для любого числа $\varepsilon > 0$ найдется число $\delta > 0$ такое, что для каждой точки $x \in I$ выполняется $\|f - \pi\|_{X_p[J_{x,h}]} < \varepsilon \cdot h^{k+\frac{1}{p}}$ при $0 < h < \delta$, $J_{x,h} = [x-h;\ x+h] \cap I$, где π – многочлен из условия (k,p)-дифференцируемости в точке x.*

Теорема 1 ([3])**.** *Если функция f равномерно (k,p)-дифференцируема на I при некотором $p \ge 1$, то $f \in C^k[I]$, $f_p^{(k)} = f^{(k)}$ (находится в классе эквивалентных функций).*

Любая функция f из $C^k[I]$ является, конечно, равномерно (k,p)-дифференцируемой на I при всех $0 < p \le \infty$.

Изучение (k,p)-дифференцируемости функции на отрезке тесно связано с продолжением оператора k-кратного дифференцирования (Λ_p^k, $0 < p < \infty$,) первоначально определённого на пространстве C^k (см. [4]), а также разработкой численных алгоритмов исследования дифференциальных моделей.

2. Вычисление производных в пространствах L_p, $p \geq 1$

Актуальным вопросом в обсуждаемой тематике является вычисление тейлоровских производных (коэффициентов многочлена, определяющего гладкость функции).

Пусть $f \in L_1[a;b]$. Рассмотрим для $h>0$ и $x \in [a;\ b-h]$ «одностороннюю» функцию Стеклова: $S_h(f,x) \stackrel{\text{def}}{=} \frac{1}{h}\int\limits_x^{x+h} f(t)dt$. Далее, для $x \in [a;\ b-(k+1)h]$, положим

$$\Delta_h^k(f,x) \stackrel{\text{def}}{=} \sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j} S_h(f,x+jh). \tag{1}$$

Отметим, если $F(x) = \int\limits_a^x f(t)dt$, то $S_h(f,x) = \dfrac{\Delta_h^1(F,\ x)}{h}$ и

$$\Delta_h^k(f,x) = \Delta_h^k\Big(\frac{1}{h}\,\Delta_h^1(F),x\Big) = \frac{1}{h}\,\Delta_h^{k+1}(F,x),$$

где

$$\Delta_h^m(F,x) = \sum_{j=0}^{m}(-1)^{m-j}\binom{m}{j} F(x+jh), \quad m \in \mathbf{N},$$

– обычная m-я разность функции F в точке x.

Для $h<0$ в формуле (1) рассматриваются значения $x \in [a+(k+1)h;\ b]$.

Замечание. Конструкции на основе функции Стеклова применялись в пространствах L_p при построении разностных выражений и аналогов модулей гладкости (см., например, [5]), но они включают в себя и значения функции в отдельных точках, от чего в некоторых ситуациях логично отказаться. В статье [6] вычислительная конструкция на основе только $S_h(f,x)$ применялась для нахождения k-ой производной функции $f \in C^k$.

Предложение 2. *Если функция f является (k,p)-дифференцируемой, $p \geq 1$, в точке $x \in (a;b)$, то существуют*

$$\lim_{h\to 0}\frac{\Delta_h^m(f,x)}{h^m} = f_p^{(m)}(x), \quad m = 1,\ \ldots,\ k.$$

Для $x=a$ или $x=b$ рассматриваются односторонние пределы.

Доказательство. При $p > 1$, естественно, $f_p^{(m)}(x) = f_1^{(m)}(x)$. Без потери общности будем считать $x=a=0$ и рассматривать $h>0$. Из условия получаем, что выполняется

$$\Big\|f - (a_0 + \cdots + a_m t^m + \cdots + a_k t^k)\Big\|_{L_1[0;\,(k+1)h]} = o(h^{k+1}) \ \text{ при } \ h \longrightarrow 0.$$

Применяя Предложение 1 при $m = 1,\ \ldots,\ k$, запишем
$f(t) = a_0 + \cdots + a_m t^m + \gamma_m(t)$, где функция γ_m такова, что
$\int\limits_0^{(m+1)\cdot u}\Big|\gamma_m(t)\Big|dt = o(u^{m+1})$, $u>0$. Получаем

$$\frac{\Delta_h^m(f,0)}{h^m} = \frac{\Delta_h^{m+1}(F,0)}{h^{m+1}} = \frac{1}{h^{m+1}}\,\Delta_h^{m+1}\Big(a_0 u + \cdots + a_m\frac{u^{m+1}}{m+1} + \int\limits_0^u \gamma_m(t)dt,\ 0\Big).$$

Для конечных разностей хорошо известно соотношение (см., например, [7], с. 159 или [8], с. 54):

$$\Delta_h^{m+1}(u^j, x) = (m+1)!\ h^{m+1}\ \delta_{j,\,m+1}, \quad j = 0,\ 1,\ \ldots,\ m+1,$$

где $\delta_{j,i}$ – символ Кронекера. Если $\Gamma_m(u) \stackrel{\text{def}}{=} \int\limits_0^u \gamma_m(t)dt = o\Big(\big(\frac{u}{m+1}\big)^{m+1}\Big)$, то $\Delta_h^{m+1}\big(\Gamma_m, 0\big) = o(h^{m+1})$. Что влечёт

$$\lim_{h\to 0} \frac{\Delta_h^m(f,0)}{h^m} = m!\ a_m = f_p^{(m)}(0).$$

□

По заданной функции $f \in L_1[a;b]$ и разбиению $\tau_n = \{[x_{i-1}; x_i)\}_{i=1}^n$ полуинтервала $[a;b)$ на равные полуинтервалы построим ступенчатую функцию, определяемую формулой

$$\Lambda_n^m[f](x) = \frac{\Delta_h^m(f,\ x_{i-1})}{h^m} \quad \text{при } x \in [x_{i-1}; x_i)$$

(заключительный справа полуинтервал замыкаем), где $h = \frac{b-a}{n(m+1)}$.

Теорема 2. *Пусть $p \ge 1$. Если функция f является (k,p)-дифференцируемой в точке $x \in [a;b]$, то в этой точке последовательность $\big\{\Lambda_n^m[f]\big\}$ при $n \to \infty$ сходится к $f_p^{(m)}(x)$, $m = 1,\ \ldots,\ k$. Если f равномерно (k,p)-дифференцируема на $[a;b]$, то $f \in C^k$ и $\big\{\Lambda_n^m[f]\big\}$ сходится равномерно к $f^{(m)}$, $m = 1,\ \ldots,\ k$.*

Доказательство. По условию для $m = 1,\ \ldots,\ k$ имеем

$$f(t) = a_0(x) + a_1(x)\cdot(t-x) + \cdots + a_m(x)\cdot(t-x)^m + \gamma_m(x,t),$$

где функция γ_m такова, что $\big\|\gamma_m\big\|_{L_1[J_{x,u}]} = o(u^{m+1})$ для $J_{x,u} = [x-(m+1)\cdot u;\ x+(m+1)\cdot u] \cap [a;b]$, $u > 0$.

При доказательстве первого утверждения без потери общности будем считать, что $a < 0$, $b > 0$, $x = 0$. Для краткости вместо $\gamma_m(0,t)$ пишем $\gamma_m(t)$. Рассмотрим некоторое разбиение τ_n отрезка $[a;b]$. Пусть полуинтервал $J_i = [x_{i-1}; x_i)$, $1 \le i \le n$, (при $i = n$ – отрезок $[x_{n-1}; x_n]$) из τ_n содержит точку 0. Получаем

$$\Big|\Lambda_n^m[f](0) - f_p^{(m)}(0)\Big| = \left|\frac{\Delta_h^m\left(a_m t^m + \gamma_m(t)\right),\ x_{i-1})}{h^m} - m!\ a_m\right| = \left|\frac{\Delta_h^m\left(\gamma_m(t)\right),\ x_{i-1})}{h^m}\right|.$$

Пусть $(m+1)\cdot u = \max\{-x_{i-1},\ x_i\}$, тогда $u \le h \le 2u$ и, следовательно (см. Предложение 2),

$$\Delta_h^m\left(\gamma_m(t)\right), x_{i-1}) = o(h^m).$$

Поскольку условие $n \to \infty$ влечёт $h \to 0$, то получаем нужное утверждение.

Если f равномерно (k,p)-дифференцируема на отрезке, из Теоремы 1 следует $f \in C^k$. Поэтому $f_p^{(m)} = f^{(m)}$, $m = 1,\ \ldots,\ k$, и равномерно по x выполняется $\big\|\gamma_m(x,t)\big\|_{L_1[J_{x,t-x}]} = o((t-x)^{m+1})$, значит, $\big\{\Lambda_n^m[f]\big\}$ сходится равномерно к $f^{(m)}$. □

Следствие 1. *Если $f \in W_1^k$, то $\big\{\Lambda_n^k[f]\big\} \xrightarrow{\text{п.в.}} f^{(k)}$ при $n \to \infty$.*

Теорема 3. *Если $f \in W_p^k$, то $\left\{\Lambda_n^k[f]\right\}$ сходится в пространстве L_p.*

Доказательство. Оценим сначала в условиях теоремы $\left\|\Lambda_n^k[f]\right\|_p$ при произвольном значении $n \in \mathbf{N}$. По определению,

$$\left\|\Lambda_n^k[f]\right\|_p = \left(\sum_{i=1}^{n} \left|\frac{\Delta_h^k(f,\ x_{i-1})}{h^k}\right|^p (k+1)h\right)^{\frac{1}{p}} = (k+1)^{\frac{1}{p}} \left(\sum_{i=1}^{n} \left|\frac{\Delta_h^{k+1}(F,\ x_{i-1})}{h^{k+1}}\right|^p h\right)^{\frac{1}{p}},$$

где $F(x) = \int\limits_a^x f(t)dt$. Для $(k+1)$-й разности функции из W_1^{k+1} выполняется (см. [8], с. 137)

$$\Delta_h^{k+1}(F, x) = h^{k+1} \int\limits_0^{(k+1)h} \frac{N_{k+1}(t/h)}{h}\ F^{(k+1)}(x+t)\ dt,$$

где N_{k+1} – нормализованный B-сплайн порядка $k+1$ с узлами в точках $0,\ \ 1, \cdots, k+1$. То есть N_{k+1} – неотрицательная кусочно-полиномиальная функция степени k, принадлежащая пространству $C^{k-1}(-\infty; \infty)$, имеющая носитель $(0;\ k{+}1)$ с $k{+}2$ равноотстоящими узлами на нём (включая конечные точки интервала) и обладающая свойством $\int\limits_0^{k+1} N_{k+1}(t)\ dt = 1$ ([8], с. 128, формула (4.40)). Кроме того, $\|N_{k+1}\|_{B[0;\ k+1]} \le 1$ ([8], с. 125, формула (4.32)).

Рассмотрим отдельно случай $p = 1$. Очевидно

$$\left\|\Lambda_n^k[f]\right\|_1 = (k{+}1) \sum_{i=1}^{n} \left| \int\limits_0^{(k+1)h} N_{k+1}(t/h)\ f^{(k)}(x_{i-1}+t)\ dt \right| \le (k{+}1) \cdot \left\|f^{(k)}\right\|_1.$$

Пусть $1 < p < \infty$. Тогда по интегральному неравенству Гёльдера

$$\begin{aligned} \left\|\Lambda_n^k[f]\right\|_p &\le (k+1)^{\frac{1}{p}} \left(\sum\nolimits_{i=1}^{n} \left(\int\limits_0^{(k+1)h} \left(\frac{N_{k+1}(t/h)}{h}\right)^{\frac{p}{p-1}} dt\right)^{p-1} \left(\int\limits_0^{(k+1)h} \left|f^{(k)}(x_{i-1}+t)\right|^p dt\right) h\right)^{\frac{1}{p}} \\ &= (k+1)^{\frac{1}{p}} \left(\int\limits_0^{(k+1)h} \left(N_{k+1}(t/h)\right)^{\frac{p}{p-1}} d(t/h)\right)^{\frac{p-1}{p}} \left(\sum\nolimits_{i=1}^{n} \int\limits_{x_{i-1}}^{x_i} \left|f^{(k)}(t)\right|^p dt\right)^{\frac{1}{p}} \le \\ &\le (k+1)^{\frac{1}{p}} \cdot \left\|f^{(k)}\right\|_p. \end{aligned} \tag{2}$$

Заключительное неравенство в преобразованиях основано на том, что из свойств $\|N_{k+1}\|_{B[0;\ k+1]} \le 1$ и $\|N_{k+1}\|_{L_1[0;\ k+1]} = 1$ вытекает $\|N_{k+1}\|_{L_q[0;\ k+1]} \le 1$ при любом $1 < q < \infty$.

Для заданных функции $f \in W_p^k$ и $\varepsilon > 0$ найдётся функция $f_\varepsilon \in C^k$ такая, что $\left\|f^{(k)} - f_\varepsilon^{(k)}\right\|_p < \varepsilon$. Получаем, что в неравенстве

$$\left\|\Lambda_n^k[f] - f^{(k)}\right\|_p \le \left\|\Lambda_n^k[f] - \Lambda_n^k[f_\varepsilon]\right\|_p + \left\|\Lambda_n^k[f_\varepsilon] - f_\varepsilon^{(k)}\right\|_p + \left\|f^{(k)} - f_\varepsilon^{(k)}\right\|_p$$

первое и третье слагаемые в правой части могут быть сделаны одновременно достаточно малыми за счёт выбора функции f_ε (неравенство (2)), а второе слагаемое станет меньше любой требуемой величины, начиная с некоторого соответственно большого номера n. □

References

[1] S. Bernstein, "On the Question of Local Best Approximation of Functions", in *Dokl. USSR Acad. Sci*, vol. 26, 1940, pp. 839–842.

[2] A. Calderón and A. Zygmund, "Local properties of solutions of elliptic partial differential equations", in *Selected Papers of Antoni Zygmund*, Springer, 1989, pp. 285–339.

[3] A. Morozov, "On Taylor Differentiability in the Spaces L_p, $0 < p \leq \infty$", *Modeling and analysis of inform. systems*, vol. 25, no. 3, pp. 323–330, 2018.

[4] A. Morozov, "Local approximations of differentiable functions.", *Math. Notes*, vol. 100, no. 2, pp. 256–262, 2016.

[5] V. Abilov and F. Abilova, "Problems in the approximation of 2π-periodic functions by Fourier sums in the space $L_2(2\pi)$", *Math. Notes*, vol. 76, pp. 749–757, 2004.

[6] A. Khromov and G. Khromova, "Discontinuous Steklov operators in the problem of uniform approximation of derivatives on an interval", *Computational Mathematics and Mathematical Physics*, vol. 54, no. 9, pp. 1389–1394, 2014.

[7] V. Dzyadyk, *Introduction to the theory of uniform approximation of functions by polynomials*. Nauka, 1977.

[8] L. Schumaker, *Spline Functions: Basic Theory*. Wiley, New York, 1981.