ISSN 1818–1015 (Print)
ISSN 2313–5417 (Online)

Министерство науки и высшего образования Российской Федерации
Ярославский государственный университет им. П. Г. Демидова

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

Том 26 2 (80) 2019

Основан в 1999 году
Выходит 4 раза в год

Адрес редакции: ЯрГУ, ул. Советская, 14, г. Ярославль, 150003, Россия
Website: http://mais-journal.ru, e-mail: mais@uniyar.ac.ru; телефон (4852) 79-77-73

Научные статьи в журнал принимаются по электронной почте. Статьи должны содержать УДК, аннотации на русском и английском языках и сопровождаться набором текста в редакторе LaTeX.

12+

СОДЕРЖАНИЕ

Моделирование и анализ информационных систем. Т. 26, №2. 2019

Computer system organization

Theory of data

Theory of computing

Computing methodologies and applications

Algorithms

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. Учредитель – федеральное государственное бюджетное образовательное учреждение высшего образования "Ярославский государственный университет им. П. Г. Демидова". Подписной индекс – 31907 в Объединенном каталоге "Пресса России". Компьют. верстка М.С. Каряевой. Редактор перевода Э.И. Соколова. Подписано в печать 17.06.2019. Дата выхода в свет 30.06.2019. Формат 60x84$^1/_8$. Усл. печ. л. 15,6. Уч.-изд. л. 6,6. Объем 122 с. Тираж 46 экз. Свободная цена. Заказ 029/019. Адрес типографии: ул. Советская, 14, оф. 109, г. Ярославль, 150003 Россия. Адрес издателя: Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

ISSN 1818–1015 (Print)
ISSN 2313–5417 (Online)

P.G. Demidov Yaroslavl State University

MODELING AND ANALYSIS OF INFORMATION SYSTEMS

Volume 26 No 2 (80) 2019

Founded in 1999
4 issues per year

Editorial Office Address: P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia
Website: http://mais-journal.ru, e-mail: mais@uniyar.ac.ru

Contents

Modeling and Analysis of Information Systems. Vol. 26, No 2. 2019

Computer system organization

Theory of data

Theory of computing

Computing methodologies and applications

Algorithms

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 195–202
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 195–202

Computer system organization

DOI: 10.18255/1818-1015-2019-2-195-202

UDC 004.415.538

Threshold Analysis of Request Degradation in the Computer Network

Kirnos V. P.

Received August 10, 2018
Revised May 13, 2019
Accepted May 15, 2019

Abstract. In this paper, the existing approaches to the assessment of computer networks performance are considered. The standard structure of a network of the application layer of the OSI model using the example of *SBIS3* application (product of Tensor Company) is treated.

Further, two approaches allowing to analyze degradations in a network are considered – on the basis of aggregated data and the operational analysis.

The degradation study of more than 60 000 request types between two versions of application which works on the basis of the computer network is the cornerstone of the first decision. Each type of requests is described by four based metrics, each metrics representing a time series. The input data are aggregated every 10 minutes before an analysis algorithm. Further, the threshold criteria based on mathematical expectation and dispersion within two adjacent versions of the software are used. Such an approach allows to significantly reduce time for the analysis of potential problems in case of updates within the computer network.

The second decision is based on not aggregated input data. It consists of detail information about all requests, there are data section of the computer network. A threshold criterion is based on durations in the selected queue. This analysis type allows to diagnose the errors with problem clients.

Keywords: threshold criteria, mean, metric, queue

For citation: Kirnos V. P., "Threshold Analysis of Request Degradation in the Computer Network", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 195–202.

On the authors:
Vasilii P. Kirnos, orcid.org/0000-0002-6150-3352, QA performance engineer
Tensor Company
36 Uglichskaya str., Yaroslavl 150027, Russia, e-mail: vp.kirnos@tensor.ru

Introduction

The paper studies the problem of continuous delivery. It shows how we promote our software into production more and more effectively. Every version release candidate is treated as a real productive version. Our understanding of how the Web works has led us to develop better ways of having machine-to-machine communication. Virtualization platforms allowed us to provide and resize our machines at will, with infrastructure automation giving us a way to handle these machines at scale. The main solution is setting Service Level Agreement (SLA) for testing the system [2].

1. Infrastructure

In our company we divided all functionality in the service architecture. There are more than 700 services in our structure: 100 of them are services for users, 50 are for all cloud administration and 550 of other types. Services are not the smallest unit. In the world development practice you can find that the concept of microservices is more adaptive. To transform a services-oriented structure to a microservices-oriented architecture is not easy [1].

Every month we normally have a new code of all and new services in production with its own logic with machine to machine communication. Our top management does not like when the data center grows after the release iteration. That is the main issue for quality assurance. Every month we have to control the resources which are used by the code, that is rather complicated as there are more than 700 developers.

2. Logs in the System

All these services have a logging system, which is used by the developers to optimise their functionality. We use these logs to make issues in our troubleshooting system. But the data from the logs are so great, that nobody can easily analyze them string by string.

We decided that we would save the operation logs (Fig. 1) only for maximum 3 days, and if we want more, we need some aggregation. The aggregation is based on

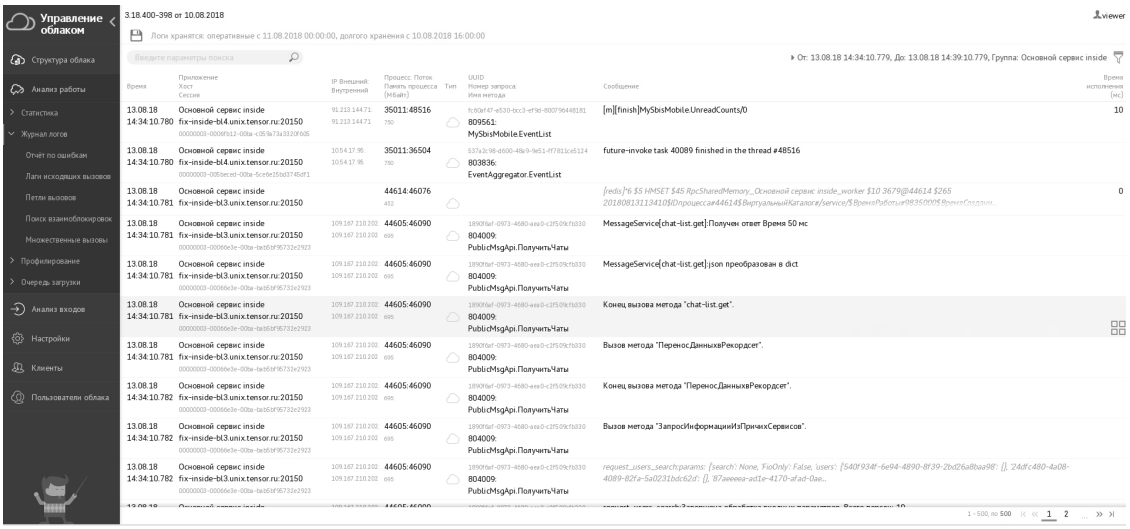


Fig 1. Operation logs

ClickHouse system (Fig. 2) developed by the Yandex team. It is a very fast column-oriented database for preparing the report. There are 6 stands in the company: two of them are only for developers, two for testing, one for debugging production database with a new middleware code and a pre-released stand.

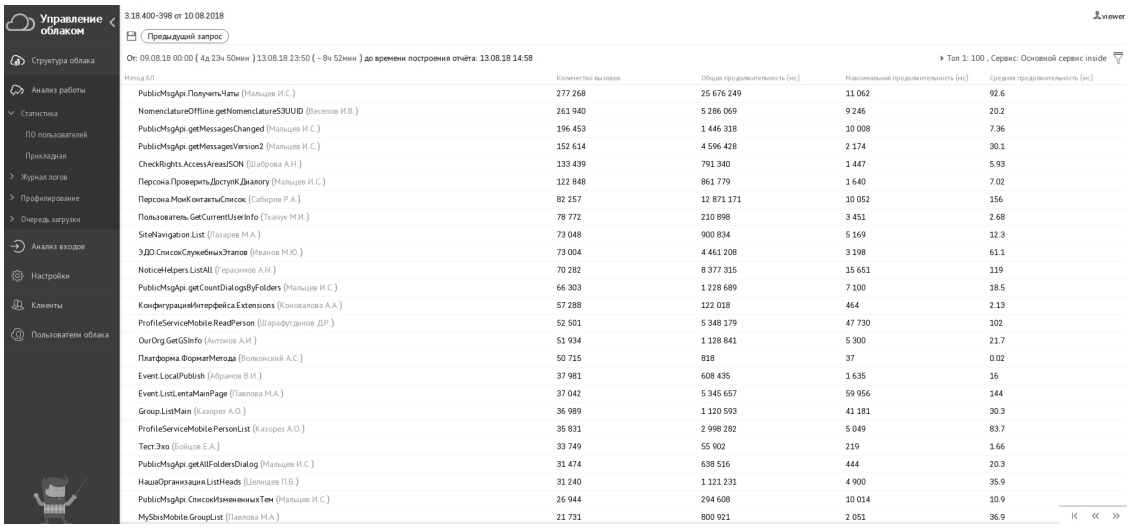


Fig 2. Aggregated logs

3. Aggregated Data Method of Analysis

The protocol of M2M communication in the cloud is json-rpc over http by default. Each service has its own method to communicate with the others. There are more than 60 000 methods of business logic in the cloud.

We are trying to compare methods performance between two major releases. There are almost more than 60 metrics for comparison. But for us the main metrics are Average time, Maximum time, Quantity of calls and Summarized time. All methods can be visualized and after that analyzed by engineers of the performance testing department, but it usually takes 4 seconds to prepare one figure and 2 seconds to analyze it in a good situation.

Preparing and analyzing 60 000 will take more than 83 hours of human time. That is why we try to automate such process.

3.1. Data Preparation

We send a post request to the special service in the cloud, which works with the ClickHouse database and can prepare the data in readable form for users.

In our case, it is the most difficult work because we use the functionality that was prepared for table visualization, not for high frequency calls. The bottleneck is Clickhouse: each request to the service should return to us a file in csv-format with 10 methods and 4 metrics with 10 minute detalization.

3.2. Data Analysis

Python3 is used to analyze data [3]. The guideline for the solution is based on an expected value (1) and dispersion (2). The expected value (3) must change not more than 50 percent from version to version, and dispersion (4) – not more than 100 percent. For the

test environment the dispersion is not indicative.

$$M(X) = \sum_{i=1}^{n} x_i/i, \tag{1}$$

$$\sigma^2(X) = \frac{\sum_{i=1}^{n} {x_i}^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}{n}, \tag{2}$$

$$Mp(X_{v1}, X_{v2}) = \frac{\sum_{i=1}^{n_{v2}} x_i/i - \sum_{j=1}^{n_{v1}} x_j/j}{\sum_{j=1}^{n_{v1}} x_j/j} \cdot 100\%, \tag{3}$$

$$\sigma^2(X_{v1}, X_{v2}) = \left(\frac{\sum_{i=1}^{n_{v2}} {x_i}^2 - \frac{(\sum_{i=1}^{n_{v2}} x_i)^2}{n}}{\sum_{j=1}^{n_{v1}} {x_j}^2 - \frac{(\sum_{j=1}^{n_{v1}} x_j)^2}{n_{v1}} \frac{n_{v1}}{n_{v2}}} - 1 \right) \cdot 100\%, \tag{4}$$

After automatic analysis we have more than 3 000 suspicious methods metrics.

There is a trouble connected with the unstability of the test stand. Every time our developers want to fix some bugs on the stands, this fix sometimes destroys the stand. We added an alerting database with a SLA trigger for stands. After getting the data from the logging system and the data of a SLA trigger, we processed the data by a logical AND.

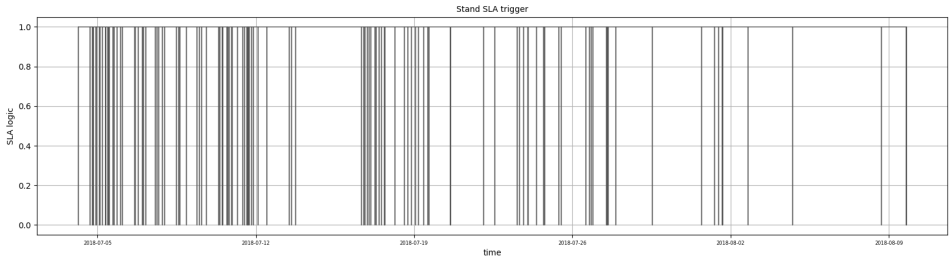


Fig 3. SLA trigger

3.3. Supervision Control

Not all metrics are needed for issue in the bugtracker system. We use supervision analysis over the automatic one. A quality assessment engineer checks all the suspicious results after the automatic script. There are also some troubles. At the production we have seasonal subseries, on the test stands they are not so seasonal. There are periodical peaks on the images. These peaks are the results of the system update. For better supervision control we divide all values into percentiles, and plot images of the metric fluctuation lower than 99th percentile (Fig. 4).

3.4. The Statistics Results

The table 1 shows the result by using the performance analysis.

In version1 we did not use the algorithm on the test stand, and there were a lot of performance bugs in the production. Version2 was the first attempt without the SLA database and the percentile division. Version3 is the current result.

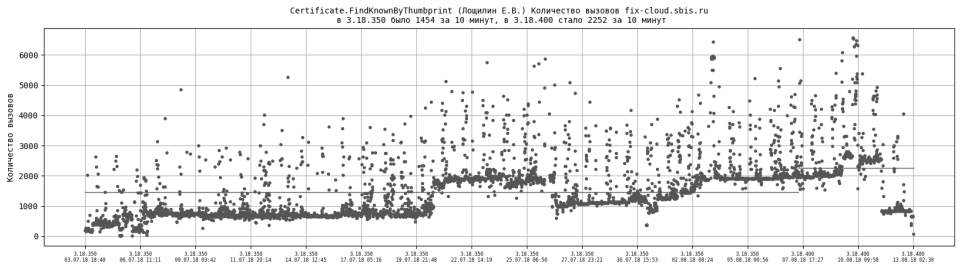


Fig 4. Degradation example

Table 1. Errors per version

version/stand	test *suspicious metrics / issues in bugtracker*	pre-release *suspicious metrics/ issues in bugtracker*	production *suspicious metrics / issues in bugtracker*
version1	-	-	4 218/18
version2	4 571 / 25	3 213 / 15	3 568 / 10
version3	5 764 / 34	3 319 / 10	3 211 / 2

4. Operations Log Analysis

This type of logs is not very convenient to analyze [4, 5]. The easiest way is to make some statistics with small step, less than 10 minutes. That is why we decided to prepare our statistics by the ClickHouse database based method.

4.1. The Queues

Our system is complicated, and for communication we used queues. The queues are infinite: we can add a lot of requests. There are three types of queues:

- **The main queue**.This queue is very important for customers. It is not permissible if a request is executed more than 2 seconds on production and 4 seconds on the test stands.
- **The service queue**. It is a queue that is not very important to customers. It was made for M2M communication.
- **The async queue**. This is an asynchronous queue. It is a part of the service queue, but it uses not all the processes in the service pool of the server's application (the service pool is light grey in Fig. 5 and the main pool is dark). This queue was made for long requests.

4.2. The Main Metrics in Pools

There are three main metrics which we analyze in the queue:

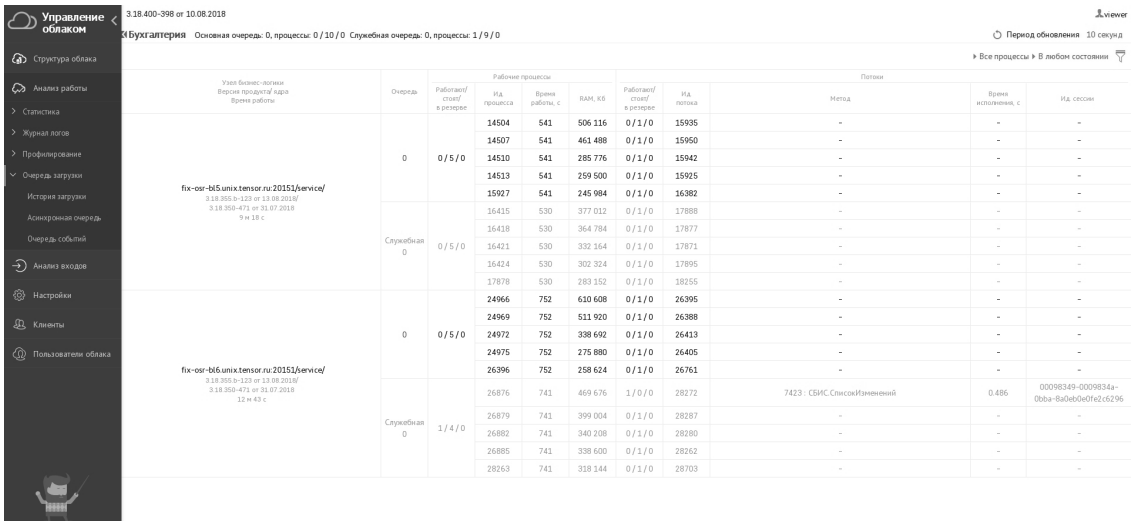


Fig 5. Example of workers pools

- **Waiting in the queue**.This time request is waiting until the worker in the pool is free for execution (Fig. 6).
- **The queue length**. This metrics shows us how many requests are in the queue (Fig. 7).
- **Execution time**. The main metrics of the request.
- **All time in the cloud**. This time is the sum of the previous two. The customer can normally see this time in the browser when the browser is building the page.

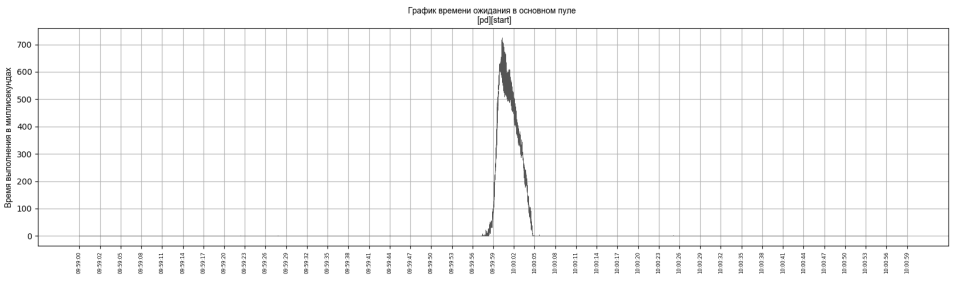


Fig 6. Example of workers pools

All those metrics help us to make a decision what type of request has become the queue reason.

4.3. Queue Analysis

The level for starting queue analysis is 250 ms on the production and 4 000 ms on the test environment.

To begin our analysis, firstly we build the top on the summarize execution time (Fig. 8). After that we use correlation between each method in the top and the waiting “func-

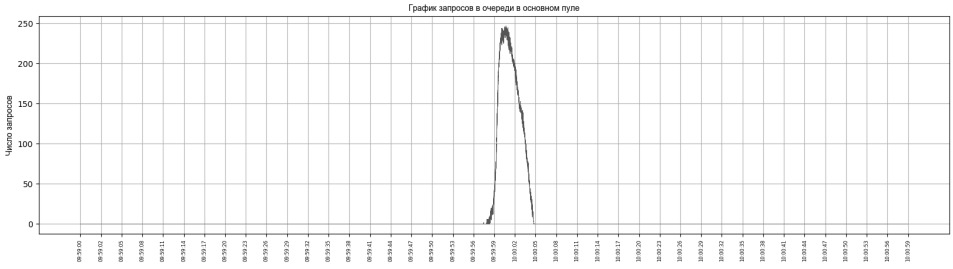


Fig 7. Example of workers pools

tion" (Fig. 6). In 95% the trouble type of requests is in the top-5, but sometimes these types of request are rather fast and they can be out of the top.

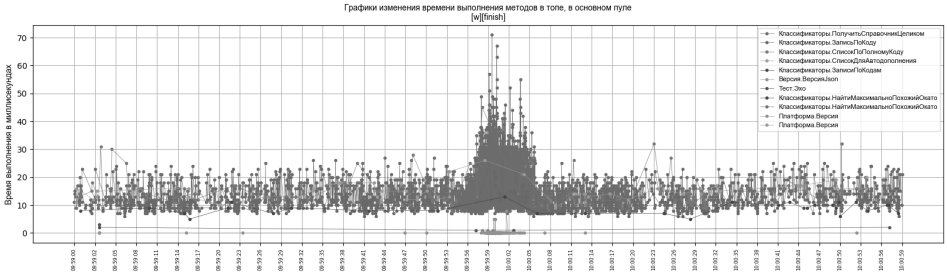


Fig 8. Top-10 of the summarize execution time

The statistics by graphics for some QA engineers in our company is not the easiest way to describe an error to bugtracker, that is why we prefer to add statistics in the table for the correct decision. Some troubles in the system are connected with the actions of some users. We use statistics by ip-address to detect such users and to predict user's scenario of the bug case.

5. Conclusion and Future Ideas

Here are two solutions that we use to detect troubles in our company. There are still a lot of issues need to be solved in future:

- **The statistics analyzer**. The current aggregated solution gives a lot of suspicious metrics of methods back to the supervisor. It is not very convenient to write a bug. We need some experiments with other statistic criteria to make better automatic decisions.
- **The queue analyzer**. The developers are not satisfied, because we could not give them the correct user case to reproduce the bug with the queue on the service.

References

[1] Newman S., *Building Microservices: Designing Fine-Grained Systems*, O'Reilly Media, New York, 2015.

[2] Wieder P., Butler J. M., Theilmann W., Yahyapour R., *Service Level Agreements for Cloud Computing*, Springer Publishing Company, Incorporated, 2011.

[3] Irdis I., *Python Data Analysis*, Packt Publishing, 2014.

[4] Anand M., "Cloud monitor: Monitoring applications in cloud.", *IEEE Cloud Computing for Emerging Markets, CCEM 2012 - Proceedings*, 2012, 58–61.

[5] Khazaei H., Fokaefs M., Zareian S., Beigi-Mohammadi N., Ramprasad B., Shtern M., Gaikwad P., and Litoiu M., "How do I choose the right NoSQL solution? A comprehensive theoretical and experimental survey", *Big Data and Information Analytics (BDIA)*, **(2):1** (2016).

Кирнос В. П., "Пороговый анализ деградации запросов внутри вычислительной сети", *Моделирование и анализ информационных систем*, **26**:2 (2019), 195–202.

DOI: 10.18255/1818-1015-2019-2-195-202

Аннотация. В данной работе рассматриваются существующие подходы к оценке производительности вычислительных сетей. Представлена типовая структура сети прикладного уровня модели OSI на примере приложения *СБИС3* (продукт Компании Тензор). Далее рассмотрены два подхода, позволяющие анализировать деградации внутри сети на основе агрегированных данных и оперативного анализа.

В основе первого решения лежит исследование деградации более 60 000 типов запросов между двумя соседними версиями приложения, которое работает на базе вычислительной сети. Каждый тип запросов описывается четырьмя основными метриками, каждая метрика представляет собой временной ряд. На вход алгоритму анализа поступают агрегированные данные выборками по 10 минут. Далее используются пороговые критерии, основанные на математическом ожидании и дисперсии в рамках двух соседних версий программного обеспечения. Такой подход позволяет существенно сократить время для анализа потенциальных проблем при обновлениях в вычислительной сети.

Информация о каждом запросе на том или ином участке вычислительной сети служит входными данными для второго решения. В качестве порогового критерия выбирается продолжительность ожидания в очереди. Этот тип анализа позволяет диагностировать дефекты, которые становятся причиной δ-образных очередей продолжительностью от нескольких секунд до 10 минут. Подобные дефекты практически не диагностируются в рамках первого подхода.

Ключевые слова: пороговый критерий, матожидание, метрика, очередь

Об авторах:
Кирнос Василий Павлович, orcid.org/0000-0002-6150-3352, программист-тестировщик
ООО Компания "Тензор"
ул. Угличская, 36, г. Ярославль, 150027 Россия, e-mail: vp.kirnos@tensor.ru

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 203–212
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 203–212

DOI: 10.18255/1818-1015-2019-2-203-212

UDC 517.9

On the Way to SD-WAN Solution

Korsakov S., Sokolov V.

Received April 29, 2018
Revised May 21, 2019
Accepted May 23, 2019

Abstract. The article describes a background and some steps in the implementation of an industrial solution to build manageable mesh overlay network on top of a complete or partially non-manageable underlay network. The overlay network has (or may have) some features from the software defined networks world. We call this solution as SD-WAN Lite to highlight that this solution is not (and will not be in a visible future) a complete SD-WAN solution.

Keywords: software defined networks, mesh networks, routing protocol, network node, encrypted data connection, generic router encapsulation protocol, multi-point GRE tunnel

For citation: Korsakov S., Sokolov V., "On the Way to SD-WAN Solution", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 203–212.

On the authors:
Stanislav Valentinovich Korsakov, orcid.org/0000-0003-2349-7980, CEO,
NETSHe Lab Ltd,
Belinskogo str., 28-75, Yaroslavl, 150047, Russia,
Assistant Professor,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl, 150003, Russia e-mail: sta@stasoft.net

Valery Anatolyevich Sokolov, orcid.org/0000-0003-1427-4937, Doctor, Professor,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl, 150003, Russia, e-mail: valery-sokolov@yandex.ru

Acknowledgments:
This work was supported by RFBR under the Grant №17-07-00823-a.

Introduction

In this article, we use the following agreement on terms:
BGP – Border gateway protocol. The routing protocol and software suite;
OSPF – Open Shortest Path First.The routing protocol and software suite;
HUB – a network node which accepts initial data connections and plays the major role in the routing process;
SPOKE – a network node which initiates data connections;
CONTROLLER – a network node which authorizes other nodes in the network, forms and deploys the node configuration;
IPSec – a method and a protocol set to establish encrypted data connections;
GRE – a generic router encapsulation protocol;

mGRE – a multi-point GRE tunnel;
DM VPN – a method to establish an overlay network with the use of mGRE tunnels, IPSec and some routing engines on the top of mGRE tunnels. DM VPN operates with at least one HUB and one or more SPOKEs;
SDN – software defined networks;
SD-WAN – a kind of methods to establish an overlay network through classical networks (non SDN) with some features taken from SDN;
TE – traffic engineering. A set of methods to manage traffic flows in the network;
VPN – Virtual private network;
NHRP – Next hop resolution protocol;
NAT – Native address translation;
NBMA – Non broadcasting multiply access networks.

The article describes a concept of the solution of the task how to establish a manageable overlay network on the top of non-manageable public IP networks and methods to provide security and TE in such networks.

1. Common Requirements for SD-WAN Lite Solution and Design Stages

At the first stage of the project, we analyzed some trends, existing and announced solutions in the target area. As a result, we have formed some mandatory requirements for implementation. The solution shall:

- be simple in configuration;
- provide a zero touch (or near to zero touch) deploy;
- provide a fault tolerant overlay network configuration;
- establish a mesh network (as HUB-SPOKE connections as well as SPOKE-SPOKE connections) without manual intervention;
- have automatic exchange of routing information;
- support one or two links to the controller and (or) HUBs. The configuration with two links must support a hot link backup;
- support secured data connections and transfer only;
- be able to manage traffic flows between nodes (to find best path, to have backups, to be able to do traffic engineering).

The second stage of the project included search and analysis for existing technologies and methods which can be used in the solution as a proof of concepts, partially or completely. Priority was provided for standard and widely used technologies which have open source implementations at this step. We assumed that the last thing can reduce time and required resources to implement the final solution.

During the second stage, we selected two technologies to use in the final solution:

– DM VPN as a primary method to establish the overlay network and to secure data [3].

– OSPF as the primary routing method because it has traffic engineering features inside.

The selected things match common requirements and allow to establish the secured mesh overlay network on top of the common IP network.

As we may remark, such a secured mesh network is the same as VPN.

The selected things defined a typical solution topology. It includes at least one controller, two or more HUBs and at least one SPOKE. The controller processes SPOKE and HUB requests, authorizes requests, helps to select a reliable access method and data, verify the right LAN network assignment and deploy the result settings to SPOKEs and HUBs. HUB acts as a key point in a virtual private network organization and routing and as a part of traffic engineering process in the future. SPOKE initiates VPN organization, acts as a source for routing and traffic engineering data.

2. DM VPN Overview

Why is DM VPN selected as a basement for SD-WAN Lite solution?

DM VPN is the technology to establish secured VPN over public IP networks such as the Internet.

DM VPN relies on three proven, widely used and standard technologies:

1. The Next Hop Resolution Protocol (NHRP); it creates a distributed (NHRP) mapping database of all the spoke tunnels to real (public interface) addresses [1, 2].

2. The multi-point GRE Tunnel Interface – a single GRE interface to support multiple GRE tunnels.

3. IPSec; it secures data through GRE tunnels.

To simplify and automate the routing management, DM VPN also uses a dynamic routing service like OSPF or BGP.

DM VPN operates with two types of nodes:

– HUB, which represents a central office or a cloud service and controls all VPN connections;

– SPOKE, which represents a branch office and establishes a connection with the central office and (may be) another branches.

DM VPN can implement as star (all traffic in VPN goes through HUB) as well as full mesh (the traffic goes between branches) topologies.

It is possible to prohibit spoke-to-spoke communications by firewall or advanced routing rules in the star topology. This case may be useful to meet different enterprise requirements.

Mesh topology allows to unload hub channels, to reduce the hub load and to reduce the spoke-to-spoke delay.

Spokes may have statically or dynamically assigned IP address, may be behind NAT [1].

The hub must have a reachable IP address and must not be behind NAT [2].

The hub may be addressed through FQDN (Full Qualified Domain Name). In case of FQDN addressing, the use of dynamically assigned IP addresses is allowed with understanding and agreement about service interruption when Hub address changes.

2.1. DM VPN Components. NHRP

NHRP provides registration, resolution and redirect services [1, 2].

NHRP registration:

– a spoke dynamically registers its mapping with NHS;

– supports spokes with dynamic NBMA addresses or NAT.

NHRP resolutions and redirects:

– support building dynamic spoke-to-hub and spoke-to-spoke tunnels;

– create star or full-mesh overlay network topology.

2.2. DM VPN Components. Multi-point GRE Tunnel

It provides single tunnel interface and NHRP source. Single tunnel interface (multi-point):

– a Non-Broadcast Multi-Access (NBMA) network;

– a smaller hub configuration;

– a multicast and broadcast support;

– a dynamic tunnel destination.

Next Hop Resolution Protocol (NHRP):

– VPN IP-to-NBMA IP address mapping;

– short-cut forwarding;

– direct support for dynamic addresses and NAT.

[1] Some types of NAT or NAT settings may prohibit DM VPN Spoke functionality.

[2] The hub may be behind properly configured NAT with multi-point GRE and IPSec bypassing

2.3. DM VPN Components. IPSec

DM VPN builds out a dynamic tunnel overlay network.

IPSec is triggered through "tunnel protection" and works together with NHRP:

- NHRP triggers IPsec before installing new mappings;
- IPSec notifies NHRP when the encryption is ready;
- NHRP installs mappings and sends registration if needed;
- NHRP and IPSec notify each other when a mapping or service assurance is cleared.

2.4. DM VPN Components. Dynamic Routing

The dynamic routing service such as BGP or OSPF is used to establish automatically right routing rules to Hub and every Spoke.

Each available routing service has its own advantages:

- OSPF requires a bit less configuration in simple cases. It allows to use traffic engineering features.
- BGP provides less delay for the overlay network reconfiguration and routing exchange.

2.5. Major Features

DM VPN offers a configuration reduction and no-touch deployment:

- it supports IP Unicast, IP Multicast and dynamic routing protocols;
- it supports remote peers with dynamically assigned addresses;
- it supports spoke routers behind the dynamic NAT and hub routers behind the static NAT.

It has open source NHRP implementation for Linux-based system (OpenNHRP and port of OpenNHRP for Quagga) [3].

Dynamic spoke-to-spoke tunnels for scaling partial- or full-mesh VPNs use IPsec encryption to secure data.

2.6. Typical DM VPN Use Cases [3]

Controlled corporate extranet network.
Star topology. Spoke-to-spoke communications are prohibited (or controlled) by firewall or routing rules.

Meshed corporate network.
Meshed topology. Spoke-to-spoke communications are allowed to reduce latency and hub load.

DM VPN as the backup for a MPLS network.
Meshed topology. Spoke-to-spoke communications are allowed to reduce latency and hub load. DM VPN is used only when the primary MPLS network is down.

[3] Both implementations do not support authentication and NBMA.

3. Traffic Engineering Elements in the Solution

BGP does not have TE elements and was rejected as routing engine for solution. In case of OSPF, some trivial traffic engineering elements exist 'out of box':

- best path selection;
- route injection;
- interface cost and priority [7];
- route (HUB) metrics;
- different areas.

Also, OSPF has more TE elements like bandwidth and delay based to implement more reliable traffic management in the future [4]-[6]. Such a development must be based on a non-manageable entity of an underlay network and on a big delays/slow feedback between nodes in an overlay network. We think that this area mostly requires statistical methods and is perspective for further study.

As a result of development, we have got solution (software) which is based on Linux, implements DM VPN as primary overlay method and OSPF as primary routing method.

3.1. DM VPN Implementation Details and Limitations in SD WAN Lite

Current solution provides:

- simplifield (few steps) configuration;
- Hub and Spoke functionality out of the box;
- PSK protected IPSec tunnels;
- OSPF or BGP routing over the overlay network. Routing does not require any special configuration.

At the same time, we have some limitations:

- we support only dual hub configuration for fault-tolerance;
- we have compatibility with CISCO Hub and Spoke only. Another vendor Hub and Spoke compatibility with OSPF has not been tested yet.

3.2. SD WAN Lite Implementation Background

To get the solution, we used some existing software like Linux kernel as OS basement, Quagga as BGP and OSPF routing daemon, OpenNHRP port to Quagga as NHRP daemon; StrongSWAN as IPSec management daemon.

In the development process, we encountered some issues in Linux-based DM VPN implementation:

1. Multicast packets could not forward through the mGRE tunnel in Linux. It disallows to run OSPF through mGRE tunnels.

2. NHRP implementation does not support Cisco-style authentication.

3. NHRP implementation works only with host networks (/32 or /128).

4. OSPF implementation in Quagga cannot run over unconnected networks (Through /32 networks. According to the RFCs). It disallows the distribution of OSPF hello and update packets through mGRE tunnels.

We have made some patches for Quagga OSPF and NHRP daemons to resolve this issues.

For example, we implemented Cisco-style authentication for Quagga NHRP (issue 2) and made patch to Quagga OSPF to allow working through interfaces with /32 (for Ipv4) and /128 (for Ipv6) masks (issue 4).

To complete OSPF support in DM VPN, we rewrote tunnel support in Linux kernel to fix multicast packet transfer through multipoint tunnels (issue 1).

The patch code does lookup in routing and neighbor tables to find valid outer address for every multi-point tunnel endpoints and to send a fixed copy of the original packet with a discovered outer address to every endpoint.

3.3. DM VPN Cons as a Method to Build an Overlay Network

As DM VPN has its own pros, as well as its own cons. And the main con is a trouble (up to full impossibility) to traverse some NAT and firewalls.

E.g. it will not work in case when IPSec is prohibited by carrier or will have troubles in mesh topology for SPOKE with low traffic.

To overcome this issue, we must add a component which meets the next requirements:

– it is able to traverse most of NAT types and firewalls;

– it is able to provide similar security for data transfer as IPSec;

– it is able to run the same routing engine as DM VPN part.

3.4. Alternative to DM VPN Method to Build Overlay Networks

We have found that OpenVPN in TCP mode can be successfully used to traverse most of firewalls and NATs which are not traversal by DM VPN (especially with using port 443).

OpenVPN provides strong encrypted connections and security for data transfer through these connections is similar to IPSec.

OpenVPN can be used with OSPF or BGP as an overlay routing engine.

Depending on these facts, OpenVPN can be used as the second method to establish an overlay network in SD-WAN Lite solution in addition to DM VPN.

The OpenVPN place in the solution:

– when SPOKE sends a registration request to the controller, it reports properties of SPOKE Internet connections;

– the controller checks which connections do not have a real IP address and (or) exist behind the NAT;

– after checking, the controller assigns to use DM VPN and starts to measure connection time. If SPOKE does not establish DM VPN connection in some time, it requests HUB again and gets settings to establish OpenVPN connection as the last resort.

3.5. A Big Fat Con Against OpenVPN Method

The use of OpenVPN implements the standard star-like topology between HUB and SPOKE and reduces an area for any kinds of traffic engineering.

We have spoke about CONTROLLER above. Let us explain place and role for CONTROLLER in the solution below.

CONTROLLER has relations with HUB(s) as well as with SPOKE(s). Such relations may be represented as charts.

3.5.1. HUB-Controller Interactions Chart

– HUB sends registration requests to the controller;

– the controller validates requests, forms and deploys HUB configuration;

– HUB reports traffic and routing properties to the controller;

– the controller forms management actions for HUB (affects QoS and routing).

3.5.2. SPOKE-Controller Iterations Chart

– SPOKE sends registration requests to the controller;

– the controller validates requests, forms and deploys SPOKE configuration;

– SPOKE reports link, traffic and routing properties to the controller;

– the controller forms management actions for SPOKE (affects QoS and routing. For example, through the interface cost and (or) TE metrics);

– the controller forms management actions for HUB (affects routing).

3.5.3. Base Solution Security

Initial transactions between a SPOKE candidate and the controller and a HUB candidate and the controller go encrypted over HTTPS with the mandatory certificate validation.

All transactions and data transfer between HUB and SPOKE and SPOKE and SPOKE go over IPSec encrypted tunnels or OpenVPN encrypted tunnels.

Security key points are:

– a key to access the controller;

– passphrases and (or) certificates which are deployed from the controller;

– NHRP secrets which are deployed from the controller;

– a root certificate which is used to sign the controller certificate and which must be kept in every HUB and SPOKE.

Physical access to any device provides ability to stole the root certificate, implements MITM attack and obtain any credentials to connect a rogue device in the network and provides some destructive actions in a future.

3.6. Security Enhancement

To improve security, the controller will change and redistribute secrets, passphrases and (or) node certificates by some algorithm (for example, by schedule).

To minimize losses in case of a rogue device, the solution must have:

– a feature to divide SPOKEs to "trusted" and "untrusted". Only trusted SPOKEs can establish meshed connections, can have full routing updates, can submit traffic engineering data, can be involved in the certificate and credentials exchanging process. Untrusted SPOKEs can only establish SPOKE-to-HUB connections, can have aggregated routes only to minimize a potential destructive effect. Their data will be ignored in traffic engineering and management processes. SPOKE state exchange is performed manually;

– a feature to upload a new root certificate to all nodes and to regenerate any inherited certificates;

– a set of features to detect possible rogue SPOKEs;

– a feature to allow console/WebUI access to SPOKE after the central user authentication (for example, against RADIUS server in the controller). Some failed logins will mark SPOKE as a possible rogue device;

– route announce validation. We assume that non-rogue SPOKE will announce non-overlapped networks and this network set will be stable in the time, will not try to forward any kind of traffic through the device, will not try to establish 'strange' routes.

4. Conclusion

When writing the article, we have got a proof of the concept for SD-WAN Lite solution where:

– the complete Cisco-style DM VPN solution is working and well tested;

– the Open VPN TCP mode part is working and well tested;

– OSPF-based routing with some trivial traffic engineering elements is working;

– the proof of the concept for the solution controller is obtained.

References

[1] Luciani J., Katz D., Piscitello D., Cole B., Doraswamy N., "RFC 2332: NBMA Next Hop Resolution Protocol (NHRP)", *IETF, April*, 1998, https://tools.ietf.org/html/rfc2332.

[2] Fox B., Petri B., "RFC 2735: NHRP Support for Virtual Private Networks", *Request for Comments,'Online*, 1999, https://tools.ietf.org/html/rfc2735.

[3] *Cisco. Dynamic Multipoint VPN Configuration guide.*, https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/xe-16/sec-conn-dmvpn-xe-16-book.html.

[4] Katz D.,Kompella K., Yeung D., "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2", *Internet RFCs*, **1** (2003), 1, https://tools.ietf.org/html/rfc3630.

[5] Giacalone S., Ward D., Drake J., Atlas A., Previdi S., "RFC-7471. OSPF Traffic Engineering (TE) Metric Extensions", 2015, https://tools.ietf.org/html/rfc7471.

[6] Retvari G., Cinkler T., "Practical OSPF traffic engineering", *IEEE Communications Letters*, **8**:11 (2004), 689-691.

[7] Nemeth K., Korosi A., Retvari G., "Optimal OSPF traffic engineering using legacy Equal Cost Multipath load balancing", *2013 IFIP Networking Conference. – IEEE*, 2013, 1-9, https://www.researchgate.net/publication/261208279_Optimal_OSPF_traffic_engineering_using_legacy_Equal_Cost_Multipath_load_balancing.

Корсаков С. В., Соколов В. А., "На пути к SD-WAN решению", *Моделирование и анализ информационных систем*, **26**:2 (2019), 203–212.

DOI: 10.18255/1818-1015-2019-2-203-212

Аннотация. В статье описываются предпосылки и некоторые этапы реализации промышленного решения по построению управляемой ячеистой оверлейной сети поверх полностью или частично неуправляемой опорной сети. Оверлейная сеть имеет (или может иметь) некоторые функции из мира программно-определяемых сетей. Мы называем это решение SD-WAN Lite, чтобы подчеркнуть, что это решение не является (и не будет в обозримом будущем) полным SD-WAN решением.

Ключевые слова: программно-определяемые сети, ячеистые сети, протокол маршрутизации, сетевой узел, зашифрованное соединение данных, общий протокол инкапсуляции маршрутизатора, многоточечный GRE туннель

Об авторах:
Станислав Валентинович Корсаков, orcid.org/0000-0003-2349-7980, директор,
ООО «Нетше лаб»,
ул. Белинского, 28-75, г. Ярославль, 15004,7 Россия,
ассистент кафедры теоретической информатики,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003, Россия, e-mail: sta@stasoft.net

Валерий Анатольевич Соколов, orcid.org/0000-0003-1427-4937, докт. физ.-мат. наук., профессор, заведующий кафедрой теоретической информатики,
Ярославский государственный университет им. П.Г. Демидова, ул. Советская, 14, г. Ярославль, 150003, Россия, e-mail: valery-sokolov@yandex.ru

Благодарности:
Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта №17-07-00823-а.

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 213–228
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 213–228

Theory of data

DOI: 10.18255/1818-1015-2019-2-213-228

УДК 517.9

Об обнаружении атак типа повторного использования исполнимого кода

Косолапов Ю. В.

Поступила в редакцию 17 декабря 2018
После доработки 13 мая 2019
Принята к публикации 15 мая 2019

Аннотация. При эксплуатации уязвимостей программного обеспечения типа переполнения буфера в настоящее время часто используется техника повторного использования кода. Такие атаки позволяют обходить защиту от исполнения кода в стеке, реализуемую на программно-аппаратном уровне в современных информационных системах. В основе атак лежит нахождение в уязвимой программе подходящих участков исполнимого кода – гаджетов – и сцепление этих гаджетов в цепочки. В статье предлагается способ защиты приложений от атак, использующих повторное использование кода. Способ основан на выделении свойств, которые позволяют отличить цепочки гаджетов от типичных цепочек легальных базовых блоков программы. Появление во время выполнения программы нетипичной цепочки базовых блоков может свидетельствовать о выполнении вредоносного кода. Одним из свойств цепочки гаджетов является исполнение в конце цепочки специальной инструкции процессора, используемой для вызова функции операционной системы. Для операционной системы Linux на базе архитектуры x86/64 проведены эксперименты, показывающие важность этого свойства при выявлении исполнения вредоносного кода. Разработан алгоритм выявления нетипичных цепочек, который позволяет выявлять все известные на настоящий момент техники повторного использования кода.

Ключевые слова: повторное использование исполнимого кода, уязвимости программного обеспечения

Для цитирования: Косолапов Ю. В., "Об обнаружении атак атак типа повторного использования исполнимого кода", *Моделирование и анализ информационных систем*, **26**:2 (2019), 213–228.

Об авторах:
Косолапов Юрий Владимирович, orcid.org/0000-0002-1491-524X, канд. техн. наук
Южный Федеральный Университет,
ул. Мильчакова, 8а, г. Ростов-на-Дону, 344090 Россия, e-mail: itaim@mail.ru

Введение

В информационных системах, построенных на базе архитектуры x86/64, с целью защиты от вредоносного кода, эксплуатирующего уязвимости типа переполнения буфера (для защиты от эксплоитов), используется техника защиты от выполнения программного кода в стеке и защита от записи в страницы с кодом. Эта техника получила название защита типа «запись, либо исполнение» (write xor execute

$W \oplus X$). Однако этот защитный механизм может быть обойден при использовании техники возвратно ориентированного программирования (ROP – return oriented programming) [1,2]. В основе техники ROP лежит логика инструкции `retq`, которой обычно завершаются подпрограммы (функции) машинного кода для архитектуры x86/64: из ячейки памяти, на которую указывает регистр `rsp` (указатель стека), в регистр-указатель инструкций `rip` загружается адрес следующей выполняемой инструкции, при этом указатель `rsp` увеличивается на 8 байтов (для 64-разрядного процессора). При использовании техники ROP в стек во время эксплуатации уязвимости типа переполнения буфера помещается не сам исполнимый код эксплоита, а адреса и аргументы для *гаджетов* — небольших участков кода, уже находящихся в области памяти уязвимой программы с разрешенным исполнением программного кода и заканчивающихся инструкцией `retq`. В этом случае эксплоит — это последовательность гаджетов (ROP-гаджетов), также называемая ROP-цепочкой. На рис. 1. приведен пример использования возвратно ориентированного программирования. В примере предполагается, что в ходе переполнения буфера удается перезаписать значения стека таким образом, что перед возвратом из уязвимой функции (перед выполнением инструкции `retq` этой функции) стек имеет вид, показанный на рис. 1 слева. Тогда при выполнении инструкции `retq` в регистр `rip` загрузится значение ячейки, на которую указывает регистр `rsp`. Таким образом передается управление по адресу `address1`. В результате выполнится код первого гаджета: в регистр `rbp` загрузится значение `val` из стека (инструкция `popq %rbx`) и выполнится инструкция `retq` этого гаджета. При выполнении этой инструкции регистр `rsp` указывает на ячейку со значением `address2`, поэтому, следуя логике исполнения инструкии `retq`, передается управление на второй гаджет по адресу `address2`. Таким образом, указанное в левой части рисунка содержимое стека приводит к тому, что к текущему содержимому регистра `rax` прибавляется значение `val` и результат помещается в регистр `rcx`.

	Содержимое стека
	...
rsp⇒	address1
	val
	address2
	...

Адрес гаджета	**Код гаджета**
address1	popq %rbx retq

address2	addq %rax,%rbx
	movq %rbx,%rcx
	retq

Рис. 1: Пример содержимого стека (слева) и области исполнимой памяти (справа) при использовании возвратно ориентированного программирования
Fig. 1: An example of stack contents (at the left) and executable memory area (at the right) when using return oriented programming

Задача разработчика эксплоита с использованием техники ROP состоит в нахождении необходимых гаджетов в адресном пространстве программы, содержащей уязвимость. Так как большинство программ используют функции стандартных библиотек, загружаемых в область памяти создаваемых процессов, то часто поиск ROP-гаджетов осуществляется в машинном коде этих библиотек [1], [2]. Например, в стандартной библиотеке GNU libc.so.6 для архитектуры x86/64 насчитывается до

17 550 гаджетов [3], с помощью которых может быть реализован любой алгоритм (в этом случае говорят, что набор гаджетов обладает полнотой по Тьюрингу). Так как до появления технологии ASLR (address space layout randomization) стандартные библиотеки загружались в память процесса по фиксированному адресу, то использование возвратно ориентированного программирования позволяло достаточно просто обходить защиту типа $W \oplus X$. Технология ASLR препятствует включению в эксплоит конкретных значений адресов гаджетов. Однако уязвимости, приводящие к утечке информации об адресах, в совокупности с наличием уязвимостей типа переполнения буфера позволяют в ряде случаев обходить защиту ASLR. Также для обхода такой защиты используется техника распыления кучи (heap spray) [4].

Для защиты от ROP-цепочек предложены различные механизмы: теневой стек [5] (реализуется путем внесения дополнительных инструкций процессора), контроль графа потока управления [6], [7] (реализуется на аппаратном уровне, либо на уровне компиляции программ), марковские цепи [8] (выявление необфусцированных ROP-цепочек в документах), шифрование адреса возврата перед помещением его в стек [9] (однако не все компиляторы следуют правилу, согласно которому инструкции `callq` и `retq` должны находиться в строгом соответствии), подсчет числа неверно угаданных предсказаний [10] (реализуется путем внесения дополнительных инструкций процессора), подсчет числа подряд идущих небольших наборов команд, заканчивающихся инструкцией `retq` [11] (реализуется путем внесения изменений в функциональность процессора). Отметим, что компанией Intel разработана технология CET (Control-flow Enforcement Technology) [12] для защиты от атак, изменяющих поток управления инструкций. В частности, в CET реализован теневой стек и добавлены инструкции для работы с ним. Заметим, что перечисленные способы защиты не используют сигнатуры атак, поэтому позволяют обнаруживать неизвестные атаки.

Разновидностями техники ROP являются такие техники, как JOP (jump oriented programming) [13] и COP/PCOP (call oriented programming/ pure call oriented programming) [14]. Одним из отличий этих техник от ROP является то, что гаджеты заканчиваются не инструкций `retq`, а, соответственно, инструкциями косвенного перехода `jmpq` и `callq`, при этом в JOP и COP используются как гаджеты, заканчивающиеся инструкцией `retq`, так и гаджеты с завершающими инструкциями `jmpq` и `callq`. В PCOP используются только гаджеты, заканчивающиеся инструкцией `callq`. В техниках JOP/COP/PCOP необходимо наличие управляющего гаджета (dispatcher), который загружает адреса выполняемых гаджетов и на который передается управление после выполнения очередного гаджета, либо необходимо наличие наборов связующих инструкций (например, в JOP таким набором может быть последовательность инструкций `popq reg; jmpq reg`, которая реализует логику инструкции `retq`). Новые техники позволяют обходить методы защиты, разработанные для техники ROP. Например, защита на основе теневого стека или на основе подсчета числа подряд идущих инструкций `retq` не позволяет выявлять эксплоиты, построенные на основе JOP/COP/PCOP. В [15] для защиты от техники JOP используется подсчет расстояния (количество инструкций) между адресом инструкции `jmpq/callq` и адресом целевого перехода. Однако этот способ защиты не выявляет короткие цепочки гаджетов. Обобщенный метод защиты из [11], основанный на подсчете числа подряд идущих небольших наборов команд, заканчивающихся инструкцией косвенного перехода, также может быть нивелирован, если в цепочку

гаджетов вставлять специальные гаджеты, которые расцениваются системой защиты как нормальные гаджеты [16], [17] (например, гаджеты большого размера).

В настоящей работе ставится задача на основе подсчета числа подряд встречающихся инструкций косвенного перехода среди инструкций передачи управления разработать способ выявления эксплоитов, основанных на использовании техник ROP/JOP/COP/PCOP, устойчивый к включению маскирующих гаджетов. Задача решается на примере операционной системы семейства Linux для процессора с архитектурой x86/64, однако он может быть применим и для других операционных систем для архитектур x86/32 и x86/64.

Статья состоит из введения, двух разделов и заключения. В первом разделе выделяются отличительные свойства цепочек гаджетов от цепочек базовых блоков программы и строится алгоритм выявления цепочек гаджетов. Во втором разделе приводятся результаты эксперимента по оценке порогового значения одного параметра разработанного алгоритма. В заключении рассматриваются ограничения разработанного способа и некоторые возможные варианты его дальнейшего усовершенствования.

1. Алгоритм выявления атак повторного использования кода

Рассмотрим информационно-аналитическую модель $\mathcal{M} = (S, O, P)$, состоящую из программы S, входными данными которой является документ O. Программа P является системой защиты, целью которой служит блокирование выполнения программы S, когда на вход поступает документ O, содержащий вредоносный код, эксплуатирующий уязвимость в программе S. Такой вредоносный код далее будем называть эксплоитом.

1.1. Цепочки базовых блоков

Обозначим символом $\mathcal{J}$ множество возможных инструкций передачи управления. К таким инструкциям относятся инструкции условного перехода (с префиксом `jaq`, `jeq` и т.п.), безусловного перехода (с префиксом `jmpq`), инструкции вызова подпрограмм (с префиксом `callq`) и инструкции возврата из подпрограмм (`retq` и т.п.). В частности, для архитектуры x86/64 в J входит инструкция вызова функции ядра операционной системы. Эту инструкцию обозначим `syscall`.

Последовательность машинных инструкций, выполняемых процессором при работе программы S с входными данными O обозначим $\texttt{Path}(S, O)$. Для $O \neq O'$ в общем случае $\texttt{Path}(S, O) \neq \texttt{Path}(S, O')$. Последовательность $\texttt{Path}(S, O)$ можно представить в виде связанной цепи *базовых блоков*:

$$B_1, B_2, ..., B_n, \tag{1}$$

где базовый блок B_i представляет собой последовательность машинных инструкций:

$$B_i = (\texttt{op}_{i,1}, ..., \texttt{op}_{i,k_i}), \tag{2}$$

в которых среди первых $k_i - 1$ нет инструкций из $\mathcal{J}$, а последняя инструкция принадлежит $\mathcal{J}$. Отметим, что группировка инструкций в базовые блоки вида (2) используется, например, при разработке компиляторов [18]. Далее, если инструкция `op` содержится среди инструкций базового блока B, будем писать $\texttt{op} \in B$. Под объединением блоков B_i и B_j будем понимать блок $B_i \parallel B_j$ вида:

$$B_i \parallel B_j = (\texttt{op}_{i,1}, ..., \texttt{op}_{i,k_i}, \texttt{op}_{j,1}, ..., \texttt{op}_{j,k_j}). \tag{3}$$

Цепь (1) преобразуем в соответствии со следующим правилом: блок, заканчивающийся инструкцией `syscall`, объединяется в соответствии с (3) со следующим блоком в цепи (если таковой имеется). Процедура преобразования выполняется, пока в цепи имеется хотя бы один, не являющийся последним, блок, который завершается инструкцией `syscall`. В результате получим цепь вида:

$$B'_1, B'_2, ..., B'_{n'}, \tag{4}$$

где $n' \leq n$. Множество всех возможных базовых блоков, из которых состоят цепочки вида (4), обозначим $\mathcal{B}$.

1.2. Свойства цепочек гаджетов

Выделим свойства цепочек гаджетов эксплоитов, которые позволят отличить их от цепочек базовых блоков (4). С одной стороны, совокупность этих свойств должна позволять снизить вероятность распознавания легальной цепочки как цепочки гаджетов, а с другой стороны, вероятность написания эксплоита, не характеризующегося этими свойствами, должна быть минимизирована. Выделяются следующие свойства:

G1) каждый гаджет завершается инструкцией с префиксом из множества

$$\mathcal{E} = \{\texttt{retq}, \texttt{jmpq}, \texttt{callq}, \texttt{syscall}\}; \tag{5}$$

G2) если гаджет в цепочке завершается инструкцией с префиксом `jmpq` или `callq`, то эти инструкции для архитектуры x86/64 в качестве операнда содержат регистр из множества

$$\mathcal{R} = \{\texttt{rax}, \texttt{rbx}, \texttt{rcx}, \texttt{rdx}, \texttt{rbp}, \texttt{rsi}, \texttt{rdi}, \texttt{r8} - \texttt{r15}\};$$

G3) инструкцией `syscall` может завершаться только последний гаджет в цепочке гаджетов;

G4) хотя бы один из гаджетов цепочки содержит инструкцию `syscall`;

G5) гаджет может содержать произвольное количество инструкций.

Поясним эти свойства. Первое свойство G1 выделено в связи с тем, что в настоящее время известны способы составления ROP-, JOP-, COP- и PCOP-цепочек, т.е. цепочек гаджетов, в которых в качестве завершающих используются инструкции из $\mathcal{E}$. Свойство G2 также характерно для гаджетов, так как в этом случае имеется возможность контролировать адрес перехода. Такие инструкции называются инструкциями косвенного перехода. Отметим, что несложно отличить инструкции косвенного перехода от инструкций прямого перехода, так как в последнем случае операндом является адрес, а в первом случае операндом является регистр из множества

$\mathcal{R}$ или ячейка памяти, адресуемая таким регистром. Свойство G3 выделено в связи с тем, что инструкция `syscall` в настоящее время не используется как инструкция для связи гаджетов. Целью эксплоита обычно является запуск программы или запись/чтение файла. Эти операции связаны с обращением к системным функциям, поэтому выделяется свойство G4. Например, по данным сайта `shell-storm.org`, все представленные на сайте эксплоиты для архитектуры x86/64 содержат инструкцию `syscall`, при этом 35 из 37 эксплоитов в качестве последней инструкции содержат инструкцию `syscall` (в основном, это системный вызов `execve`). Согласно [16] и [17], защита, основанная на подсчете длины цепочки гаджетов с ограничением на максимальный размер гаджета и минимальную длину цепочки, может быть нивелирована вставкой маскирующих гаджетов (путем включения гаджетов размера, превышающего заданный в системе защиты порог), поэтому введено свойство G5.

Отметим, что при необходимости ограничение G5 можно ослабить, потребовав, чтобы гаджет содержал ограниченное количество инструкций, так как чем больше инструкций в гаджете, тем вероятнее возникновение побочного эффекта, нежелательного для эксплоита, например, перезапись содержимого используемого регистра. Например, в способе защиты из [10] сигнал о выполнении цепочки гаджетов генерируется в случае нахождения цепочки из 10-ти и более гаджетов, каждый из которых содержит не более шести инструкций. В [19] признаком выполнения цепочки гаджетов является цепочка из не менее чем 4-х гаджетов, каждый из которых имеет длину не более семи. Символами C_l и G_l обозначим соответственно пороговое значение на длину цепочки гаджетов (минимальное значение) и пороговое значение на число инструкций в гаджете (максимальное значение).

Пусть $SysC_l(E)$ — номер последнего гаджета в эксплоите E, содержащий инструкцию `syscall`, $SysC_l = \min_E\{SysC_l(E)\}$. Порядковый номер гаджета, в котором встречается инструкция `syscall`, также будем называть расстоянием от начала цепочки до *этой* инструкции `syscall`. В [14] удалось написать эксплоит, состоящий из двух гаджетов, однако пример приведен для случая, когда не используется технология ASLR. Заметим, что в [20] для обхода ASLR потребовалось написать эксплоит, состоящий из 6 гаджетов, а в [21] эксплоит содержит не менее 4-х гаджетов. Отсюда можно предположить, что значение $SysC_l$ может быть не менее четырех. Во втором разделе это предположение подтверждается экспериментально.

Таким образом, отличительными признаками цепочки гаджетов, удовлетворяющей свойствам G1–G5, от цепочки базовых блоков, также удовлетворяющей этим же свойствам, могут быть максимальная длина цепочки и порядковый номер последнего блока (гаджета) в цепочке, содержащего инструкцию `syscall`.

1.3. Модель защиты

Заметим, что все гаджеты, используемые в техниках ROP/JOP/COP/PCOP могут рассматриваться как базовые блоки, в то же время не каждый базовый блок является гаджетом: базовый блок может заканчиваться инструкцией прямого перехода типа `callq address` или инструкцией условного перехода вида `jaq address`, тогда как гаджеты с такими инструкциями в настоящее время не используются. Таким образом, если $\mathcal{G}$ — множество гаджетов, то $\mathcal{G} \subset \mathcal{B}$. Рассмотрим множество $\mathcal{B}_0 \subset \mathcal{B}$, состоящее из базовых блоков, удовлетворяющих свойствам G1,G2,G5. Оче-

видно, что $\mathcal{G} \subset \mathcal{B}_0$. Также рассмотрим множество $\mathcal{B}_0^V (\subseteq \mathcal{B}_0)$, состоящее из базовых блоков, в каждом из которых не более V инструкций. Следующее соотношение очевидно: ... $\subseteq \mathcal{B}_0^{V-1} \subseteq \mathcal{B}_0^V \subseteq \mathcal{B}_0^{V+1} \subseteq$ .... Будем полагать, что $\mathcal{B}_0^\infty = \mathcal{B}_0$. Соотношение введенных множеств изображено на рис. 2. Если вместо множества $\mathcal{B}_0$ рассматривать его подмножество $\mathcal{B}_0^V$ для некоторого $V \in \mathbb{N}$, то возможна ситуация, когда $\mathcal{G} \not\subset \mathcal{B}_0^V$. В этом случае возможна ситуация пропуска системой защиты P эксплоита при рассматриваемом методе обнаружения: в цепочку гаджетов произвольной длины достаточно вставлять маскирующие гаджеты из более чем V инструкций.

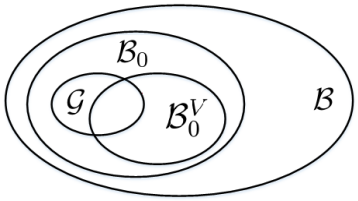


Рис. 2: Соотношение множества гаджетов и множеств базовых блоков
Fig. 2: The correspondence of the set of gadgets and sets of basic blocks

В основе предлагаемого способа защиты лежит предположение о том, что при нормальном выполнении программы S на входных данных O в последовательности (4) не появляются длинные цепочки, состоящие из базовых блоков, принадлежащих исключительно $\mathcal{B}_0$, и одновременно удовлетворяющие свойствам G3 и G4. Поэтому при справедливости этого предположения для выявления факта выполнения эксплоита предлагается в последовательности (4) выявлять подпоследовательности подряд идущих базовых блоков, каждый из которых принадлежит $\mathcal{B}_0$, и одновременно удовлетворяющие свойствам G3 и G4 с длиной более чем C_l и/или для которых последний выявленный базовый блок, содержащий инструкцию `syscall`, находится на расстоянии более чем $SysC_l$ от начала цепочки.

Для выявления цепочек гаджетов вместо цепи (4) рассмотрим последовательность, состоящую из завершающих инструкций:

$$\mathtt{op}_{1,k'_1}, ..., \mathtt{op}_{n',k'_{n'}}, ..., . \tag{6}$$

Тогда для выявления цепочек достаточно искать последовательности подряд идущих инструкций косвенного перехода (ПИИКП) в последовательности (6); префикс которых принадлежит $\mathcal{E}$, а операнд, если есть, содержит регистр из $\mathcal{R}$, причем хотя бы одна инструкция завершает базовый блок, содержащий инструкцию `syscall`. Заметим, что вычисление количества ПИИКП необходимо выполнять на основе $\mathrm{Path}(S, O)$, а не путем простого дизассемблирования соответствующего исполнимого файла программы S. Это связано с тем, что эксплоит может активизироваться только во время исполнения программы S и часто не может быть выявлен в статичном состоянии в документе O. К тому же дизассемблирование исполнимого файла может быть затруднено, если уязвимая программа защищена, например, с помощью методов обфускации.

Пусть $MChain(S, O)$ — максимальное число подряд идущих легальных базовых блоков в цепи (4), удовлетворяющих свойствам G1–G5, а C_l — как отмечалось ранее

— минимальная длина цепочки гаджетов. Если $MChain(S,O) < C_l$, то выявление исполнения цепочки гаджетов может осуществляться по правилу

$$result = \begin{cases} \texttt{exploit}, & counter > C_l \\ \texttt{typical}, & counter <= C_l \end{cases}, \tag{7}$$

где *counter* — текущая длина цепочки базовых блоков, удовлетворяющих свойствам G1–G5. Если $MChain(S,O) > C_l$, то при использовании правила (7) могут возникать ложные обнаружения цепочек гаджетов: длинная цепочка базовых блоков будет восприниматься как цепочка гаджетов. Для снижения вероятности таких событий предлагается формировать профиль программы. Под профилем программы будем понимать набор хэш-значений, полученных вычислением некоторой хэш-функции $h : \{0,1\}^* \to \{0,1\}^m (m \in \mathbb{N})$ от цепочки базовых блоков, которая встречается в Path(S, O) при выполнении программы S на легальных входных данных O, но длина которой превышает C_l. Здесь предполагается, что цепочка базовых блоков может быть записана последовательностью нулей и единиц. Если $B'_i, ..., B'_{i+r}$ — нетипичная цепочка базовых блоков длины $r + 1 (> C_l)$, в которой встречается инструкция `syscall`, то в профиль должны добавляться следующие хэш-значения $h(B'_i \parallel ... \parallel B'_{i+r-j})$, $j = 0, ..., r - C_l$. Такой профиль программы S обозначим Π_S.

Свойство G4 позволяет существенно снизить вероятность ложного срабатывания. Так, например, эксперименты, проведенные для программ FoxitReader и LibreOffice, показывают на сколько сокращается количество цепочек базовых блоков, похожих на цепочки гаджетов, а также на сколько сокращается длина цепочек. Результаты приведены в таблицах 1 и 2, где L — длины цепочки базовых блоков.

Пусть $Sys(S,O)$ — максимальное расстояние от начала цепочки базовых блоков до последнего базового блока этой цепочки, содержащего инструкцию `syscall`, при выполнении программы S на входных данных O. Если $Sys(S,O) < SysC_l$, то выявление исполнения цепочки гаджетов может осуществляться по правилу

$$result = \begin{cases} \texttt{exploit}, & last\_sys > SysC_l \\ \texttt{typical}, & last\_sys <= SysC_l \end{cases}, \tag{8}$$

где $last\_sys$ — номер последнего проверенного базового блока в текущей цепочке, удовлетворяющей свойствам G1–G5, содержащий инструкцию `syscall`. Если выполняется неравенство $Sys(S,O) > SysC_l$, то при использовании правила (8) также могут возникать ложные обнаружения цепочек гаджетов. Для снижения вероятности таких событий предлагается добавлять в профиль нетипичные легитимные цепочки, для которых $Sys(S,O) > SysC_l$. Если $B'_i, ..., B'_{i+r}$ — нетипичная цепочка базовых блоков длины $r + 1 (> SysC_l)$, в которой инструкция `syscall` последний раз встречается в блоке с номером $i + SysC_l \le t \le i + r$, то в профиль должны добавляться следующие хэш-значения $h(B'_i \parallel ... \parallel B'_{i+r-j})$, $j = r - t, ..., r - SysC_l$. Этот профиль программы обозначим Π_S^{Sys}.

На основе правил (7) и (8) построим алгоритм FindChainOfGadgets (см. алгоритм 1) выявления атак повторно использования кода. В алгоритме пороговое значение C_l используется для выявления нетипично длинных цепочек базовых блоков, удовлетворяющих свойствам G1–G5, а пороговое значение $SysC_l$ используется для выявления цепочек базовых блоков, в которых блок с инструкцией `syscall` имеет

Таблица 1: Статистика длин цепочек базовых блоков, FoxitReader
Table 1: Statistics of chain lengths of basic blocks, FoxitReader

(a) Число цепочек базовых блоков, удовлетворяющих свойствам G1–G3,G5
(a) The number of chains of basic blocks that satisfy the conditions G1–G3,G5

V	L													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
∞	4822198	425885	75179	12336	2195	1365	461	210	7	367	818	10	19	24
20	4753031	425181	74106	12101	2195	1364	462	209	7	367	818	22	9	22
15	4712871	420099	70493	12023	2241	1273	458	209	7	367	818	22	9	22
10	4506233	391030	66782	10745	2188	924	428	206	6	368	817	21	9	22
9	4392241	391112	61192	10390	2182	1058	292	205	5	366	815	21	9	22
8	4279041	290249	60446	9815	2086	816	478	10	5	366	815	21	9	22
7	3963898	247945	53062	1964	453	470	199	10	4	374	807	21	9	22
6	3818647	181893	44739	1505	631	98	198	382	3	3	807	51	0	0
5	3354795	160279	34152	982	174	95	195	2	0	0	0	12	0	0
4	2575331	100882	26002	664	27	3	0	2	0	12	0	0	0	0
3	1833687	72377	1808	433	0	1	0	0	0	0	0	0	0	0
2	965234	18570	254	1	0	0	0	0	0	0	0	0	0	0

(b) Число цепочек базовых блоков,
удовлетворяющих свойствам G1–G5
(b) The number of chains of basic blocks that
satisfy the conditions G1–G5

V	L										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
∞	167810	2550	236	47	9	8	1	2	1	3	1
20	167509	2536	227	45	9	8	1	2	1	3	1
15	167117	2476	213	41	8	8	1	2	1	3	1
10	164637	1983	153	9	6	5	1	1	1	2	0
9	163587	1847	142	7	6	5	1	1	1	2	0
8	162157	361	83	4	4	4	1	0	1	2	0
7	160648	105	21	3	3	0	0	0	1	2	0
6	159270	14	1	3	0	0	0	0	0	0	0
5	159175	9	0	0	0	0	0	0	0	0	0
4	159175	5	0	0	0	0	0	0	0	0	0
3	159174	5	0	0	0	0	0	0	0	0	0
2	145901	0	0	0	0	0	0	0	0	0	0

нетипично большой для цепочек базовых блоков порядковый номер. Параметр D_S используется для раннего выявления эксплоитов, состоящих из длинных цепочек, в которых инструкция `syscall` размещена на большом расстоянии от начала цепочки. Этот параметр индивидуален для каждой программы и может быть частью общего профиля программы S. Например, как видно из таблиц 1a и 2a, для S=FoxitReader параметр D_S может быть выбран равным 14, а для S=LibreOffice может быть выбрано значение $D_S = 7$. Параметр C_l может определяться как индивидуально для программ (в этом случае он будет частью общего профиля каждой программы), так и задаваться общим для всех программ. Например, согласно таблицам 1a и 2a, параметр C_l может быть выбран равным 6. В этом случае профиль Π_S программы LibreOffice будет пустым, а профиль программы ForitReader будет содержать

Таблица 2: Статистика длин цепочек базовых блоков, LibreOffice
Table 2: Statistics of chain lengths of basic blocks, LibreOffice

(a) Число цепочек базовых блоков, удовлетворяющих свойствам G1–G3,G5
(a) The number of chains of basic blocks that satisfy the conditions G1–G3, G5

V	L						
	1	**2**	**3**	**4**	**5**	**6**	**7**
∞	9011478	220831	189884	5503	38	5	2
20	9009423	221029	189620	5495	38	3	2
15	8964219	219859	189526	5469	37	3	2
10	8856074	214940	189370	5438	41	0	1
9	9013092	209267	78654	5425	41	0	1
8	8877798	201130	78406	5381	40	0	1
7	8846433	184202	78410	5239	20	0	1
6	4428607	152668	73059	5225	20	0	1
5	3722576	156543	70674	269	1	0	1
4	3429475	107576	66760	84	0	0	0
3	3145993	32480	60444	0	0	0	0
2	656869	177	0	0	0	0	0

(b) Число цепочек базовых блоков, удовлетворяющих свойствам G1–G5
(b) The number of chains of basic blocks that satisfy the conditions G1–G5

V	L					
	1	**2**	**3**	**4**	**5**	**6**
∞	35179	94	24	0	0	2
20	35178	94	24	0	0	2
15	33752	75	24	0	0	2
10	32989	18	0	0	0	0
9	32985	10	0	0	0	0
8	32931	10	0	0	0	0
7	32932	9	0	0	0	0
6	32932	9	0	0	0	0
5	32932	9	0	0	0	0
4	32934	5	0	0	0	0
3	32934	5	0	0	0	0
2	32376	0	0	0	0	0

только 8 нетипичных цепочек базовых блоков. Параметр V введен в алгоритм для того, чтобы имелась возможность уменьшить вероятность ложного обнаружения эксплоита, если заранее известен максимальный размер G_l гаджета, используемого в эксплоите. Если положить $G_l = 6$ (в соответствии с [10]), то как видно из таблиц 1b и 2b, возможно сокращение размера профиля (см. строки таблиц для $V \leq 6$). Если G_l заранее неизвестно, то следует использовать значение $V = \infty$.

Параметры C_l и D_S используются для раннего выявления эксплоитов, а также для выявления эксплоитов, состоящих из большого количества гаджетов, в которых не используются инструкция `syscall`. Однако в случае использования общего параметра C_l необходимо для каждой программы S хранить профиль Π_S и выполнять соответствующие проверки хэш-значений. Для исключения хранения профиля Π_S и использования правила (7) алгоритм FindChainOfGadgets может быть модифицирован так, чтобы использовалась только проверка (8). В следующем разделе проводятся эксперименты по определению параметра $SysC_l$. Профиль, состоящий из Π_S, Π_S^{Sys}, D_S и опционально из C_l может формироваться заранее, например, следующим образом: выделяется множество программ, для которых необходимо выявление эксплоитов (редакторы/просмотрщики текстовых файлов, графических файлов и т.п.), и каждая из выбранных программ запускается с входными файлами, полученными из надежных источников (без эксплоитов).

Отметим, что схожий способ защиты разработан в [22], за тем исключением, что он разработан для операционной системы Windows, в качестве гаджетов рассматриваются только те, которые завершаются инструкцией `retq` (защита только от ROP-цепочек), а также не используются профили для программ. В [22] подсчет гаджетов выполняется на основе проверки регистров LBR (Last Branch Recording): в этих регистрах сохраняются адреса возврата из функций. Если для какого-то из адресов возврата нет соответствующей инструкции `callq`, то соответствующая

команда `retq` расценивается как нетипичная. В случае появления подряд идущих нетипичных возвратов в количестве, превышающем порог, и приводящих к вызову системной API-функции, фиксируется выполнение эксплоита. Авторы [22] указывают главное ограничение своего способа: он не позволяет выявлять эксплоиты, написанные с помощью техники JOP. Также не будут выявляться цепочки, построенные с помощью техник COP/PCOP. В то же время предлагаемый в настоящей работе способ защиты от повторного использования кода позволяет выявлять цепочки гаджетов, в которых гаджеты завершаются любой инструкцией с префиксом из набора (5). Стоит отметить важное преимущество подхода из [22] – высокую скорость обнаружения эксплоитов, так как не используется эмуляция исполнения программного кода.

2. Экспериментальная оценка $SysC_l$

Для реализации разработанного способа защиты выбрана система виртуализации с открытым исходным кодом QEMU. Эта система позволяет запускать операционные системы, скомпилированные для одной архитектуры процессора (для целевой архитектуры) на процессорах другой архитектуры (на архитектуре хоста). Это достигается за счет группировки машинных инструкций для целевого процессора в базовые блоки вида (2), трансляции этих базовых блоков в промежуточный код, а затем трансляции промежуточного кода в базовые блоки, состоящие из инструкций процессора хоста. Особенностью системы QEMU является то, что описанная схема трансляции в операционных системах Linux/BSD для архитектуры x86/64 (архитектура хоста и целевая архитектура) может применяться не только для запуска операционных систем, но и для запуска приложений.

В исходные коды QEMU внесены необходимые изменения, позволяющие сохранять в отдельный файл последовательность вида (6). Для каждой инструкции сохраняется следующая информация: название инструкции, аргумент, количество машинных инструкций в базовом блоке.

Для проведения эксперимента выбраны наиболее популярные программы для просмотра pdf-файлов в операционной системе Linux Ubuntu 16.04: FoxitReader, LibreOffice, Xpdf, Okular, Evince, GNU Gv. Выбор обусловлен тем, что эксплоиты могут быть внедрены в pdf-файлы. Например, уязвимость в просмотрщике Evince [23], при открытии специально сформированного файла провести атаку типа отказа в обслуживании. Для исследования были выбраны 15 файлов формата pdf размером от 979 байтов до 1,4 МБ как с шифрованным содержимым, так и с незашифрованным. Также для сравнения полученных результатов дополнительно выбраны программы Mahjogg, Mines, Sol, Sudoku, FireFox, которые запускались без параметров. В таблице 3 для $V = \infty$ содержатся результаты нахождения $Sys(S, O)$: для всех программ в ячейках приведены максимальное и минимальное количество выявленных цепочек с заданным значением $Sys(S, O)$. Результаты для просмотрщиков pdf-файлов отделены от результатов других программ двойной линией.

Выше отмечалось, что написание коротких цепочек гаджетов, позволяющих обходить защиту ASLR, является трудной задачей. Например, в работах [20] и [21] соответствующие цепочки имеют длину не менее 4-х гаджетов. Результаты таб-

Исходные параметры: 1) Последовательность $\mathrm{Path}(S, O)$, представленная цепью базовых блоков вида (4) длины n',
2) $V \in \mathbb{N}$, 3) профиль Π_S, 4) профиль Π_S^{Sys}, 5) D_S,
6) C_l, 7) $SysC_l$.

Результат: Сообщение о наличии эксплоита (`exploit`) или отсутствии нетипичных подцепочек (`typical`)

$chain = \varnothing$, $counter = 0$, $result =$ `typical`, $syscall\_present = false$
цикл $i = 1, ..., n'$ **выполнять**
  если $B'_i \in \mathcal{B}_0^V$ **тогда**
    если `syscall` $\in B'_i$ **тогда**
      $syscall\_present = true$
    конец условия
    $counter = counter + 1$
    // Проверка длины цепочки
    если $counter > D_S$ **тогда**
      $result =$ `exploit`
      break
    конец условия
    иначе
      // Проверка (7)
      если $counter > C_l$ ***and*** $syscall\_present = true$ ***and*** $h(chain) \notin \Pi_S$ **тогда**
        $result =$ `exploit`
        break
      конец условия
      иначе
        // Проверка (8)
        если `syscall` $\in B'_i$ ***and*** $counter > SysC_l$ ***and*** $h(chain) \notin \Pi_S^{Sys}$ **тогда**
          $message =$ `exploit`
          break
        конец условия
        иначе
          $chain = chain \parallel B'_i$
        конец условия
      конец условия
    конец условия
  конец условия
  иначе
    $counter = 0$, $chain = \varnothing$, $syscall\_present = false$
  конец условия
конец цикла
возвратить $result$

Алгоритм 1: FindChainOfGadgets

лицы 3 показывают, что при $SysC_l = 4$ для большинства программ выполняется условие $Sys(S, O) < SysC_l$, и поэтому для этих программ возможно использование

только правила (8) и без составления профиля Π_S^{Sys}. В этом случае для программ FoxitReader и FireFox потребуется хранить лишь небольшие профили: из одной и восьми цепочек соответственно. Отметим, что возможно усиление защиты за счет выбора значения $SysC_l = 3$ (в открытых источниках не найдено информации об эксплоитах, состоящих из трех гаджетов и позволяющих обходить защиту ASLR). Однако в этом случае возрастают размеры профилей: для программ FoxitReader и Evince необходимо хранить соответственно три и одну цепочки, а для FireFox – не менее 30 цепочек.

Результаты показывают, что в алгоритме FindChainOfGadgets может использоваться только проверка вида (8) без использования правила (7) и без сравнения длины цепочки с D_S. Уменьшение числа проверок позволит ускорить процесс проверки файлов. Заметим, что результаты в таблице 3 согласуются с результатами работы [16], где аналогичные вычисления проводились для различных программ для операционной системы Windows. Отметим, однако, что в таблице 3 результаты получены в случае, когда системный вызов может быть любым, в то время как в [16] вычисления проводились не для всех системных вызовов, а только для защищенных, в частности для `VirtualProtect`.

Таблица 3: Распределение значений $Sys(S, O)$
Table 3: Distribution of $Sys(S, O)$

S	$Sys(S,O)$							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
FoxitReader	85063 – 170578	90 – 113	0 – 3	0 – 2	0 – 1	0	0	0
LibreOffice	16950 – 68093	57 – 73	0	0	0	0	0	0
Xpdf	123 – 425	12 – 19	0	0	0	0	0	0
Okular	6604 – 44671	56 – 105	0 – 10	0	0	0	0	0
Evince	8046 – 24716	26 – 31	0 – 1	0 – 1	0	0	0	0
GNU Gv	97 – 112	11	0	0	0	0	0	0
Mahjogg	4247	24	0	0	0	0	0	0
Mines	6807	26	0	0	0	0	0	0
Sol	4201	30	0	0	0	0	0	0
Sudoku	4824	30	0	0	0	0	0	0
FireFox	8091 – 834896	87 – 560	0 – 145	0 – 22	0 – 4	0 – 1	0 – 2	0 – 1

Заключение

Разработанный способ защиты не способен распознавать атаки повторного использования кода, в которых применяются гаджеты, завершающиеся инструкциями условного перехода с префиксами `jeq`, `jaq` и т.п. На настоящий момент в открытых источниках не найдено информации о наличии подобных эксплоитов. Также не будут распознаваться эксплоиты, в которых изменяются данные [25], так как в этих атаках могут не использоваться системные вызовы. В целом предлагаемый способ защиты не исключает написания эксплоита, не подпадающего под выделенные свойства, однако этот способ направлен на существенное затруднение написания такого кода.

Одним из существенных ограничений предлагаемой системы защиты является скорость работы, так как фактически происходит эмуляция работы процессора: за-

пуск программы с входными данными может осуществляться до нескольких минут. Однако необходимость эмуляции подтверждается исследованием, проведенным в работе [24], где показано, что статический анализ имеет ограничения в применении и не всегда позволяет обнаруживать вредоносный код. Способ может быть применим для отложенной проверки или для глубокого исследования открываемых файлов.

Отметим, что техника повторного использования программного кода применяется не только для обхода систем защиты. В последнее время эта техника рассматривается как способ создания систем защиты информации: включение цифровых водяных знаков [26], обфускация алгоритмов для защиты от исследования [27], программная стеганография [28]. Поэтому необходимо дальнейшее усовершенствование предложенного способа защиты для снижения вероятности ложного срабатывания (например, выполнение обфусцированной с помощью техники повторного использования кода программы может быть расценено как исполнение эксплоита). Разработанный способ может быть усовершенствован, например, добавлением эвристического анализа последовательности системных вызовов, использованием марковских моделей по аналогии с [8]. В частности, в предлагаемом способе в требовании G4 не накладывается ограничений на множество системных вызовов. В то же время при необходимости может быть составлен список системных вызовов, обращение к которым может свидетельствовать о наличии эксплоита. Например, появление системного вызова `clone`, `execve` после обнаруженной длинной цепочки гаджетов свидетельствуют о запуске приложения, что с большой вероятностью может свидетельствовать о выполнении эксплоита. Сократить число рассматриваемых цепочек возможно также за счет исключения из рассмотрения цепочек, представляющих собой выполнение цикла (многократное выполнение одно и того же базового блока), а также за счет исключения цепочек, в которых инструкция `syscall` встречается в первом базовом блоке.

Список литературы / References

[1] Shacham H., “The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86)”, *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, 552–561.

[2] Buchanan E., Roemer R., Shacham H., Savage S., “When good instructions go bad: generalizing return-oriented programming to risc”, *Proceedings of the 15th ACM conference on Computer and communications security*, 2008, 27–38.

[3] http://ropshell.com. Last access 26.11.2018..

[4] Binlin C., Jianming F., Zhiyi Y., “Heap Spraying Attack Detection Based on Sled Distance”, *International Journal of Digital Content Technology and its Applications(JDCTA)*, **6**:14 (2012), 379–386.

[5] Davi L., Sadeghi A., Winandy M., “ROPdefender: a detection tool to defend against return-oriented programming attacks”, *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, 2011, 40–51.

[6] Davi L., Koeberl P., Sadeghi A., “Hardware-Assisted Fine-Grained Control-Flow Integrity: Towards Efficient Protection of Embedded Systems Against Software Exploitation”, *Proceedings of the 51st Annual Design Automation Conference, San Francisco, CA, USA*, 2014, 1–6.

[7] Ge X., Talele N., Payer M., Jaeger T., "Fine-grained control-flow integrity for kernel software", *In IEEE European Symposium on Security and Privacy*, 2016, 179–194.

[8] Usui T., Ikuse T., Iwamura M., Yada T., "POSTER: Static ROP Chain Detection Based on Hidden Markov Model Considering ROP Chain Integrity", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, 1808–1810.

[9] Cawan S. C. Arnold S. R., Beattie S. M., Wagle P. M., "Pointguard: method and system for protecting programs against pointer corruption attacks", *Patent US7752459B2*, 2010.

[10] Cheng Y., Zhou Z., Miao Y., Ding X., Deng H. R., "ROPecker: A Generic and Practical Approach For Defending Against ROP Attack", *In Symposium on Network and Distributed System Security (NDSS)*, 2014, 1–14.

[11] Chen P., Xiao H., Shen X., Yin X., Mao B., Xie L., "DROP: Detecting Return-Oriented Programming Malicious Code", *Lecture Notes in Computer Science*, **5905** (2009), 163–177.

[12] "Control-flow Enforcement Technology Preview", 2017, software.intel.com/sites/default/files/managed/4d/2a/control-flow-enforcement-technology-preview.pdf. Last access 26.11.2018.

[13] Checkoway S., Davi L., Dmitrienko A., Sadeghi A. R., Shacham H., Winandy M., "Return-oriented programming without returns", *Proceedings of the 17th ACM conference on Computer and communications security*, 2010, 559–572.

[14] Sadeghi A., Niksefat S., Rostamipour M., "Pure-Call Oriented Programming (PCOP): chaining the gadgets using call instructions", *Journal of Computer Virology and Hacking Techniques*, **14**:2 (2018), 139–156.

[15] Yao F., Chen J., Venkataramani G., "Jop-alarm: Detecting jump-oriented programming-based anomalies in applications", *IEEE 31st International Conference on Computer Design (ICCD)*, 2013, 467–470.

[16] Goktas E., Athanasopoulos E., Polychronakis M., Bos H., Portokalidis G., "Size Does Matter: Why Using Gadget-Chain Length to Prevent Code-Reuse Attacks is Hard", *Proceedings of the 23rd USENIX Security Symposium*, 2014, 417–432.

[17] Carlini N., Wagner D., "ROP is still dangerous: breaking modern defenses", *SEC'14 Proceedings of the 23rd USENIX conference on Security Symposium*, 2014, 385–399.

[18] Ахо А. В., Лам М. С., Сети Р., Ульман Д. Д., *Компиляторы: принципы, технологии и инструментарий, 2-е изд.: Пер. с англ.*, М.:ООО «И.Д. Вильямс», 2008; [Aho A. V., Sethi R., Ullman J. D., *Compilers: Principles, Techniques, and Tools*, Pearson Education, Inc, 1986, (in Russian).]

[19] Kayaalp M., Schmitt T., Nomani J., Ponomarev D., Abu-Ghazaleh N., "Scrap: architecture for signature-based protection from code reuse attacks", *Proceedings of IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013)*, 2013, 258–269.

[20] https://sploitfun.wordpress.com/2015/05/08/bypassing-aslr-part-iii/. Last access 06.12.2018.

[21] Katoch V., "Bypassing ASLR/DEP.", https://www.exploit-db.com/docs/english/17914-bypassing-aslrdep.pdf. Last access 06.12.2018..

[22] Pappas V., Polychronakis M., Keromytis A. D., "Transparent ROP Exploit Mitigation Using Indirect Branch Tracing", *Proc. of the 22nd USENIX Security Symposium*, 2013, 447–462.

[23] https://www.securityfocus.com/bid/62780/info. Last access 03.12.2018..

[24] Moser A., Kruegel C., Kirda E., "Limits of Static Analysis for Malware Detection", *Proceedings of Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, 2008, 421–430.

[25] Hu H., Shinde S., Adrian S., Chua Z. L., Saxena P., Liang Z., "Data-oriented programming: On the expressiveness of non-control data attacks", *In Security and Privacy (SP) Symposium*, 2016, 969–986.

[26] Ma H., Lu K., Ma X., Zhang H., Jia C., Gao D., "Software watermarking using return-oriented programming", *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, 2015, 369–380.

[27] Gao D., "Method for obfuscation of code using return oriented programming", *Patent WO2016126206A1*, 2015.

[28] Lu K., Xiong S., Gao D., "Ropsteg: program steganography with return oriented programming", *Proceedings of the 4th ACM conference on Data and application security and privacy*, 2014, 265–272.

Kosolapov Y. V., "About Detection of Code Reuse Attacks", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 213–228.

DOI: 10.18255/1818-1015-2019-2-213-228

Abstract. When exploiting software vulnerabilities such as buffer overflows, code reuse techniques are often used today. Such attacks allow you to bypass the protection against the execution of code in the stack, which is implemented at the software and hardware level in modern information systems. At the heart of these attacks lies the detection, in the vulnerable program of suitable areas, of executable code — gadgets — and chaining these gadgets into chains. The article proposes a way to protect applications from attacks that use code reuse. For this purpose, features that distinguish the chains of gadgets from typical chains of legal basic blocks of the program are highlighted. The appearance of an atypical chain of the base block during program execution may indicate the execution of a malicious code. An algorithm for identifying atypical chains has been developed. A feature of the algorithm is that it is focused on identifying all currently known techniques of re-execution of the code. The developed algorithm is based on a modified QEMU virtualization system. One of the hallmarks of the chain of gadgets is the execution at the end of the chain of instructions of the processor used to call the function of the operating system. For the Linux operating system based on the x86/64 architecture, experiments have been conducted showing the importance of this feature in detecting the execution of the malicious code.

Keywords: code reuse, software vulnerability

On the authors:
Yury V. Kosolapov, orcid.org/0000-0002-1491-524X, PhD,
Southern Federal University,
8a Milchakova str., Rostov-on-Don 344090, Russia, e-mail: itaim@mail.ru

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 229–243
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 229–243

DOI: 10.18255/1818-1015-2019-2-229-243

УДК 517.9

Система распределения ключей на дизайнах Адамара

Деундяк В. М., Таран А. А.

Поступила в редакцию 04 апреля 2019
После доработки 20 мая 2019
Принята к публикации 22 мая 2019

Аннотация. Изучается актуальная задача распределения ключей в сообществе для обеспечения безопасности переписки между ее участниками. Для решения этой задачи могут рассматриваться системы предварительного распределения ключей в сообществе, при этом каждый пользователь получает некоторую ключевую информацию, на основе которой он затем может независимо от других участников системы вычислить необходимые общие секретные ключи для тех конференций, в которые он входит. Такие системы предварительного распределения ключей могут быть основаны на разных базовых структурах, в частности, на помехоустойчивых кодах и комбинаторных дизайнах. Слабостью подобных систем является возможность проведения коалиционных атак, когда недобросовестные пользователи системы могут объединиться в коалицию и на основе всей имеющейся у них ключевой информации попытаться вычислить общие секретные ключи других участников сообщества. Однако системой гарантируется безопасность ключей в случае, если мощность коалиции злоумышленников не превышает некоторого значения, которое определяется конструкцией системы.

В работе рассматривается разработанная нами система распределения ключей, основанная на комбинаторных дизайнах, а именно на 3-дизайнах Адамара, гарантирующая безопасность переписки пользователей при наличии коалиции из не более чем двух злоумышленников. Для исследования стойкости системы к коалиционным атакам в случае превышения предусмотренного значения мощности коалиции вводятся новые понятия комбинаторной оболочки и комбинаторного ранга подмножества кода Адамара и изучаются некоторые комбинаторные свойства кодов Адамара. Для построенной системы распределения ключей вычисляется вероятность успешного проведения коалиционной атаки на произвольную конференцию в зависимости от мощности коалиции злоумышленников.

Ключевые слова: системы распределения ключей, комбинаторные дизайны, коалиционные атаки

Для цитирования: Деундяк В. М., Таран А. А., "Система распределения ключей на дизайнах Адамара", *Моделирование и анализ информационных систем*, **26**:2 (2019), 229–243.

Об авторах:
Деундяк Владимир Михайлович, orcid.org/0000-0001-8258-2419, канд. физ.-мат. наук, доцент,
Южный Федеральный Университет,
ул. Большая Садовая, 105/42, г. Ростов-на-Дону, 344006 Россия,
ФГНУ НИИ «Спецвузавтоматика»,
пер. Газетный, 51, г. Ростов-на-Дону, 344002, Россия, e-mail: vl.deundyak@gmail.com

Таран Алексей Александрович, orcid.org/0000-0002-1357-9360, аспирант,
Южный Федеральный Университет,
ул. Большая Садовая, 105/42, г. Ростов-на-Дону, 344006 Россия, e-mail: fraktal-at@yandex.ru

Введение

В многопользовательских системах передачи данных для того, чтобы обеспечить защиту переписки группы участников этой системы, используются криптографические алгоритмы, многие из которых требуют наличия у всех членов группы общего секретного ключа [1]. Для того, чтобы получить этот ключ, могут использоваться алгоритмы выработки общего секретного ключа [2], в которых члены группы обмениваются своими открытыми ключами и на их основе вычисляют общий секретный ключ, или системы распределения ключей [3], в которых участники получают общий секретный ключ, обмениваясь данными с неким доверенным сервером. Однако в некоторых случаях обмен данными с другими членами группы или с доверенным сервером может быть невозможен или непрактичен. В указанных случаях могут использоваться системы предварительного распределения ключей [4]. В таких системах каждый участник заранее получает ключевую информацию, на основе которой он может самостоятельно вычислить с помощью оговоренного алгоритма общий секретный ключ для любой группы, участником которой он является. Простейшим примером такой системы будет система, в которой каждый пользователь получит один и тот же общий секретный ключ для общения с любым другим участником системы. Однако такая система будет неустойчива к атакам – возможный злоумышленник внутри системы будет иметь такой же ключ, как и у всех остальных пользователей, и сможет получить доступ ко всем данным, передаваемым в системе. Другим примером является система, в которой каждый пользователь получит открытые ключи всех остальных пользователей в качестве своей предварительной ключевой информации. Такая система позволит пользователям вычислить необходимые общие ключи. Но с увеличением числа пользователей данной системы, а следовательно, и числа групп, которые могут организовывать конференции, будет расти и объем предварительной ключевой информации, которую потребуется хранить каждому пользователю.

Поэтому актуальной является разработка таких систем, которые, с одной стороны, позволяют уменьшить размер предварительно распределяемой информации, а с другой стороны, не позволяют даже нескольким злоумышленникам, объединившимся в коалицию, получить доступ к переписке других пользователей. Примерами таких систем являются полилинейные системы распределения ключей [5–7], а также системы предварительного распределения ключей, построенные на основе шаблонов распределения ключей [8]. В качестве основы для построения последних могут использоваться различные комбинаторные структуры, такие как ортогональные массивы и комбинаторные дизайны [9], как это было показано, например, в [10] и [11]. Подробный обзор различных подходов для построения таких систем можно найти в [12]. Системы данного типа гарантируют безопасность ключей, при условии, что размер коалиции злоумышленников не превышает некоторого значения, определяемого конструкцией системы. Однако в случае превышения этого значения коалиция злоумышленников может получить возможность вычислить часть общих секретных ключей для переписок других пользователей. Анализ таких коалиционных атак для полилинейных систем, построенных на кодах Хэмминга и Рида–Маллера второго порядка, проведен в работах [13, 14].

Целью настоящей работы является построение и исследование стойкости системы распределения ключей, основанной на комбинаторных 3-дизайнах Адамара. В разделе 1. представлена система, которая построена на основе шаблонов распределения ключей. Полученная система гарантирует безопасность переписки пользователей при наличии коалиции из не более чем двух злоумышленников. В разделе 3. для исследования стойкости системы вычисляется вероятность успешного проведения коалиционной атаки на общие секретные ключи конференций в зависимости от степени превышения мощности коалиции злоумышленников. Эти результаты получены на основе изучения комбинаторных свойств кода Адамара, которым посвящен раздел 2.

1. Конструкция системы распределения ключей на дизайнах Адамара

Система распределения ключей строится на основе шаблонов распределения ключей, предложенных в [8]. Эти шаблоны описаны в разделе 1.1. В разделе 1.2. содержатся необходимые сведения о дизайнах Адамара, а конструкция системы представлена в разделе 1.3.

1.1. Шаблоны распределения ключей

Предположим, что имеется сообщество из n пользователей, которым требуется возможность проведения конференций. Пусть $\mathcal{U}$ – множество идентификационных номеров, каждый из которых взаимнооднозначно соответствует пользователю системы, $|\mathcal{U}| = n$. Через $\mathcal{P}$ будем обозначать множество всех возможных конференций, каждая конференция $P \in \mathcal{P}$ является подмножеством $\mathcal{U}$ и содержит идентификационные номера всех участников конференции. Рассмотрим задачу предварительного распределения ключей пользователям из множества $\mathcal{U}$ для последующего вычисления ими общих секретных ключей для конференций из $\mathcal{P}$.

Предварительным распределением ключей занимается доверенный сервер. Для этого ему понадобится следующая конструкция.

Введем множество блоков $\mathcal{B} = \{B^{(1)}, \dots B^{(\beta)}\}$, каждый блок $B^{(i)} \in \mathcal{B}$ является подмножеством $\mathcal{U}$. Каждому блоку $B^{(i)}$ сервер ставит в соответствие ключ $k^{(i)}$ из множества ключей $\mathcal{K}$. После этого распределение ключей происходит следующим образом. Все пользователи системы знают множество идентификационных номеров $\mathcal{U}$ и множество блоков $\mathcal{B}$, которые являются публичной информацией. Каждый пользователь получает от сервера ключ $k^{(i)}$, если его идентификационный номер $u^{(j)}$ находится в соответствующем блоке $B^{(i)}$. Т.к. идентификационный номер $u^{(j)}$ может одновременно принадлежать нескольким блокам $B^{(i)}$, то в качестве секретной информации у каждого пользователя будет несколько ключей $k^{(i)} \in \mathcal{K}$.

Для того, чтобы участники конференции $P \in \mathcal{P}$ могли вычислить общий секретный ключ, необходимо, чтобы $\mathcal{B}_P = \{B^{(i)} \in \mathcal{B} | P \subseteq B^{(i)}\} \neq \oslash$, т.е. чтобы существовал как минимум один блок, содержащий в себе идентификационные номера всех участников конференции P. Далее введем множество $\mathcal{K}_P = \{k^{(i)} \in \mathcal{K} | B^{(i)} \in \mathcal{B}_P\}$ общих ключей участников конференции P. Отметим, что множество $\mathcal{K}_P$ непусто, т.к. непу-

сто множество $\mathcal{B}_P$. Тогда участники конференции могут вычислить на основе $\mathcal{K}_P$ общий секретный ключ конференции с помощью публичной функции f, действующей из множества всех подмножеств $\mathcal{K}$ в некоторое ключевое пространство $\mathfrak{K}$. Ключ k_P конференции P вычисляется как:

$$k_P = f(\mathcal{K}_P). \tag{1}$$

Функция f выбирается таким образом, чтобы обеспечить совершенную секретность общего ключа (см. [7], с. 287 и [10], с. 217). В частности, общий секретный ключ можно вычислить только при условии, если имеется все множество общих ключей конференции $\mathcal{K}_P$. Например, в качестве такой функции может использоваться побитовая сумма всех общих ключей.

В сообществе предполагается наличие злоумышленников, которые могут объединяться в коалиции и обмениваться своей секретной информацией с целью получить секретные ключи конференций, участниками которых они не являются. Обозначим множество возможных коалиций в системе через $\mathcal{F}$, где каждая коалиция $F \in \mathcal{F}$ является подмножеством $\mathcal{U}$. Для того, чтобы злоумышленники не могли напрямую вычислить по формуле (1) общие секретные ключи конференций, в которые они не входят, необходимо, чтобы выполнялось следующее условие:

$$\forall P \in \mathcal{P}, \forall F \in \mathcal{F}, \{B^{(i)} | P \subseteq B^{(i)} \ \wedge \ F \cap B^{(i)} = \oslash\} \neq \oslash, \tag{2}$$

то есть существует по крайней мере один ключ $k^{(j)}$, который известен всем участникам конференции P, но неизвестен ни одному из злоумышленников из коалиции F и, следовательно, коалиция F не может напрямую вычислить общий секретный ключ K_P по формуле (1).

При выполнении условия (2) $(\mathcal{U}, \mathcal{B})$ – называется шаблоном распределения ключей для $(\mathcal{P}, \mathcal{F})$ и обозначается $(\mathcal{P}, \mathcal{F})$-KDP ($(\mathcal{P}, \mathcal{F})$ Key Distribution Pattern). В случае, когда $\mathcal{P}$ – все подмножества $\mathcal{U}$ мощности t, а $\mathcal{F}$ – все подмножества $\mathcal{U}$ мощности не более чем w, $(\mathcal{P}, \mathcal{F})$-KDP обозначается как (t, w)-KDP.

Таким образом, шаблон (t, w)-KDP может быть использован для построения системы распределения ключей, которая позволяет организовывать защищенные конференции для групп из t участников системы и при этом гарантирует безопасность всех общих секретных ключей, при условии наличия внутри системы коалиции злоумышленников размера не более чем w.

1.2. Дизайны Адамара

В качестве шаблонов распределения ключей ниже применяются комбинаторные дизайны Адамара, которые строятся с помощью симметричных матриц $2^a \times 2^a$ Адамара-Сильвестра [15]:

$$H_a = \underbrace{H_1 \otimes \ldots \otimes H_1}_{a}, \ H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Матрицу Адамара далее будем записывать в двоичном виде, т.е. заменим $1, -1$ на элементы $0, 1$ из поля $\mathbb{F}_2$ соответственно, и будем использовать прежнее обозначение H_a. Множество всех столбцов или, что тоже самое, строк матрицы H_a будем

обозначать $\mathcal{A}_a$. Множество $\mathcal{A}_a$ вложено в $\mathbb{F}_2^n$ и $|\mathcal{A}_a| = n$, где $n = 2^a$. Это множество образует $[n, a, n/2]$-код Адамара, при этом строки с номерами $2^{j-1} + 1, j = 1, \ldots, a$ образуют одну из кодовых матриц, порождающих этот код, и являются базисом в $\mathcal{A}_a$. Например,

$$H_3 = \left(\begin{array}{cccccccc} 0&0&0&0&0&0&0&0\\ 0&1&0&1&0&1&0&1\\ \hline 0&0&1&1&0&0&1&1\\ \hline 0&1&1&0&0&1&1&0\\ 0&0&0&0&1&1&1&1\\ \hline 0&1&0&1&1&0&1&0\\ 0&0&1&1&1&1&0&0\\ 0&1&1&0&1&0&0&1 \end{array}\right),$$

где подчеркнутые строки являются базисом, порождающим все остальные строки матрицы.

Напомним, что ν-(n, k, λ)-дизайном называется пара $\{\mathcal{U}, \mathcal{B}\}$, где $\mathcal{U}$ – множество из $|\mathcal{U}| = n$ элементов (точек), а $\mathcal{B} = \{B^{(1)}, \ldots, B^{(\beta)}\}$ – множество подмножеств $\mathcal{U}$, мощность каждого $|B^{(j)}| = k$, для которой выполняется условие, что любые ν элемента из $\mathcal{U}$ содержатся ровно в λ блоках из $\mathcal{B}$ [9]. В теореме Стинсона в [10], с. 223, доказано, что $(t+1)$-(n, k, λ)-дизайн является (t, w)-KDP, при

$$w < \frac{n-t}{k-t}. \tag{3}$$

По матрице Адамара H_a можно построить 3-$(2^a, 2^{a-1}, 2^{a-2} - 1)$-дизайн следующим образом: в качестве $\mathcal{U}$ берется множество номеров столбцов матрицы, а блок $B^{(2(i-1)-1+\epsilon)}$, где $i \in \{2, \ldots, 2^a\}$, $\epsilon \in \{0, 1\}$, содержит номера тех столбцов, у которых в i-той строке стоит значение ϵ (см. [9], с. 116). В этом случае из параметров дизайна и условия (3) вытекает, что $w \leq 2$. Таким образом построенный дизайн Адамара является $(2, w)$-KDP, т.е. может использоваться для построения системы для конференций с двумя участниками, гарантируя безопасность общих секретных ключей при наличии коалиции злоумышленников из не более чем двух пользователей системы.

1.3. Построение системы

Предположим, что сообществу из $n = 2^a, a > 2$ пользователей требуется система для обеспечения безопасности переписки между любой парой пользователей.

Для построения этой системы сервер генерирует матрицу Адамара–Сильвестра H_a. Каждый столбец соответствует пользователю системы. По матрице Адамара в соответствии с разделом 1.2. определяется 3-$(n, n/2, n/4 - 1)$ дизайн Адамара. Для каждого блока $B^{(2(i-1)-1+\epsilon)}$, $i \in \{2, \ldots, n\}$, $\epsilon \in \{0, 1\}$, сервер генерирует секретный ключ $k^{(2(i-1)-1+\epsilon)}$, всего $2(n-1)$ блоков и ключей. Ключи распределяются следующим образом. Пользователь j получает ключ $k^{(2(i-1)-1+\epsilon)}$, если $j \in B^{(2(i-1)-1+\epsilon)}$. Так как по свойствам дизайнов этот дизайн также является 2-$(n, n/2, n/2 - 1)$ и 1-$(n, n/2, n-1)$ дизайном, то каждый пользователь получает $n-1$ секретный ключ, по одному из каждой пары $\{k^{(2(i-1)-1)}, k^{(2(i-1))}\}$, $i \in \{2, \ldots, n\}$, а у каждой пары пользователей имеется $n/2 - 1$ общих ключей. Блоки $B^{(2(i-1)-1+\epsilon)}$ являются открытой

информацией, с их помощью пользователи могут определить, какие у них общие ключи с другими участниками системы. На основе этих общих ключей пользователи могут вычислить общий секретный ключ для обеспечения безопасности их переписки при помощи заранее оговоренного алгоритма.

Находящиеся в системе пользователи-злоумышленники могут объединяться в коалиции и, собрав имеющиеся у них секретные ключи, могут попытаться получить все общие ключи других пользователей системы. В этом объединенном множестве ключей могут оказаться все общие ключи некоторой пары пользователей, не входящих в коалицию. В этом случае злоумышленники смогут вычислить их общий секретный ключ и будут иметь возможность читать переписку этих двух пользователей. По теореме Стинсона (3) построенная выше система является $(2, 2)$-системой, то есть гарантирует безопасность переписки пары пользователей при наличии в системе коалиции из не более чем двух злоумышленников.

Для того чтобы иметь возможность прочитать переписку двух конкретных пользователей, злоумышленникам необходимо получить все $n/2 - 1$ их общих ключей. В случае превышения предусмотренной мощности коалиции может получиться так, что коалиция получит все общие ключи двух пользователей и сможет вычислить общий секретный ключ для их переписки. Следующая теорема показывает, что если коалиции удается получить все общие ключи для переписки двух пользователей, то фактически она получает и все секретные ключи по крайней мере одного пользователя.

Теорема 1. *Для того, чтобы была возможна атака на общий ключ переписки двух участников, необходимо и достаточно, чтобы коалиция злоумышленников имела все ключи по крайней мере одного участника переписки.*

Доказательство. Достаточность. Если у коалиции злоумышленников имеются все ключи одного из участников конференции, то у коалиции, очевидно, есть и все общие ключи для любой переписки с его участием.

Необходимость. Предположим противное: коалиция злоумышленников получила все общие ключи некоторой переписки двух пользователей, но при этом у нее нет полного набора ключей ни одного из участников. Это значит, что существует номер i такой, что у первого участника переписки имеется ключ $k^{(2(i-1)-1+\epsilon_i)}$, а у каждого участника коалиции – другой ключ $k^{(2(i-1)-\epsilon_i)}$, и существует номер j такой, что у второго участника переписки имеется ключ $k^{(2(j-1)-1+\epsilon_j)}$, а у каждого участника коалиции – другой ключ $k^{(2(j-1)-\epsilon_j)}$. При этом если у первого участника переписки есть ключ $k^{(2(i-1)-1+\epsilon_i)}$, то у второго должен быть ключ $k^{(2(i-1)-\epsilon_i)}$: если бы у второго также был ключ $k^{(2(i-1)-1+\epsilon_i)}$, то это бы значило, что i-тый ключ у них общий, а т.к. по условию теоремы коалиция получила все общие ключи злоумышленников, то должна была получить и этот.

Покажем теперь, что i и j – различны. Предположим, что $i = j$. Как было показано ранее, i-тый ключ не является общим для участников переписки, т.е. у одного из участников будет ключ $k^{(2(i-1)-1+\epsilon)}$, а у другого $k^{(2(i-1)-\epsilon)}$. Но т.к. каждый пользователь системы получает один из каждой пары ключей, то не может быть, чтобы коалиция не получила ни одного ключа из этой пары. Следовательно $i \neq j$.

Рассмотрим теперь подматрицу матрицы Адамара, состоящую из строк i и j и

столбцов, соответствующих участникам переписки и коалиции злоумышленников. Строка i может быть записана одним из двух следующих вариантов:

$$\begin{array}{ccc|cc} 0 & \cdots & 0 & 1 & 0 \\ 1 & \cdots & 1 & 0 & 1 \end{array},$$

а строки j:

$$\begin{array}{ccc|cc} 0 & \cdots & 0 & 0 & 1 \\ 1 & \cdots & 1 & 1 & 0 \end{array}.$$

Возьмем любой из возможных вариантов строки i и прибавим к ней любой из вариантов строки j. В результате получится одна из следующих строк:

$$\begin{array}{ccc|cc} 0 & \cdots & 0 & 1 & 1 \\ 1 & \cdots & 1 & 0 & 0 \end{array}.$$

Так как строки матрицы Адамара состоят из всех слов кода Адамара, то любая сумма строк также является словом кода и строкой матрицы. Таким образом, мы получаем строку, в которой во всех позициях соответствующих участникам коалиции 0 (1), а в позициях, соответствующих участникам переписки – 1 (0). Это означает, что злоумышленники не получили общий ключ участников переписки, соответствующей этой строке, что противоречит условию. Следовательно, предположение, что коалиция получила все общие ключи участников переписки, но при этом не получила все ключи ни одного из них – неверно.

□

Эта теорема позволяет перейти от рассмотрения коалиционной атаки на все возможные переписки к рассмотрению атак на ключи других пользователей системы.

Так как по теореме 1 атака на ключ какой-то конкретной переписки эквивалентна атаке на все ключи одного из её участников, то можно ограничиться рассмотрением только атак на ключи пользователя. Отметим, что коалиция получает все ключи пользователя, когда она может составить вектор-столбец матрицы Адамара, соответствующий этому пользователю, из своих идентификационных столбцов. Подробнее это рассматривается в модели коалиционных атак (раздел 3.) на основе изучения комбинаторных свойств кода Адамара, которым посвящен раздел 2.

2. Комбинаторные свойства столбцов матриц Адамара

Будем обозначать через b_i значение i-той координаты произвольного вектора $b \in \mathbb{F}_2^n$. Для произвольного множества $B \subset \mathbb{F}_2^n$ обозначим $B_i := \{b_i | b \in B\} \subset \mathbb{F}_2$, а для произвольного непустого множества $D \subset \mathbb{F}_2$ обозначим $I_D(B) := \{i \in \{1, \ldots, n\} | B_i = D\}$. Таким образом, для любого множества B множество номеров $\{1, \ldots, n\}$ разбивается на три непересекающихся подмножества $I_{\{0\}}(B)$, $I_{\{1\}}(B)$ и $I_{\{0,1\}}(B)$.

Будем говорить, что вектор $a \in \mathbb{F}_2^n$ комбинаторно-зависим от множества $B \subset \mathbb{F}_2^n$, если для любого $i \in \{1, \ldots, n\}$ выполняется: $a_i \in B_i$. В противном случае будем

говорить, что a не является комбинаторно-зависимым от B. Если a комбинаторно-зависим от B, т.е.

$$a \in \mathrm{desc}(B) := \{(a_1, \ldots, a_n) : a_i \in B_i\},$$

то его называют потомком множества B [16]. Будем говорить, что вектор a строго комбинаторно-зависим от B, если он комбинаторно-зависим от B, но независим от любого собственного подмножества B.

Если $B \subset A \subset \mathbb{F}_2^n$, то комбинаторной оболочкой множества B во множестве A будем называть множество $< B >_A := \mathrm{desc}(B) \cap A$. Нетрудно доказать, что

$$\mathrm{desc}(\mathrm{desc}(B)) = \mathrm{desc}(B),\ << B >_A >_A = < B >_A,\ < A >_A = A. \tag{4}$$

Множество векторов будем называть комбинаторно-независимым, если любой его элемент комбинаторно-независим от множества, состоящего из остальных векторов. Комбинаторно-независимое подмножество множества B будем называть максимальным, если не существует другого комбинаторно-независимого подмножества, в которое это множество вложено.

Следующая лемма вытекает непосредственно из определений.

Лемма 1. *Пусть $\Lambda \subset A \subset \mathbb{F}_2^n$ и $\Lambda' \subset A \subset \mathbb{F}_2^n$. Для того, чтобы $< \Lambda >_A = < \Lambda' >_A$ необходимо и достаточно, чтобы $I_D(\Lambda) = I_D(\Lambda')$ для любого непустого $D \subset \mathbb{F}_2$.*

Лемма 2. *Пусть A – линейное пространство над полем $\mathbb{F}_2$, $\Lambda \subset A$. $< \Lambda >_A$ является подпространством A тогда и только тогда, когда $I_{\{1\}}(\Lambda) = \oslash$.*

Доказательство. Пусть $\Theta = \{\theta \in F_2^n | \theta_i = 0, i \in I_{\{0\}}(\Lambda)\}$. Достаточность является следствием следующего равенства: $< \Lambda >_A = A \cap \Theta$. Так как любой вектор $c \in < \Lambda >_A$ является потомком векторов из Λ, то $c_i = 0$ для $i \in I_{\{0\}}(\Lambda)$, и, следовательно, $c \in \Theta$. С другой стороны, любой вектор из $d \in A \cap \Theta$ будет иметь $d_i = 0$ для $i \in I_{\{0\}}(\Lambda)$ и $d_j = 0$ или $d_j = 1$ для $j \in I_{\{0,1\}}(\Lambda)$, и, следовательно, является потомком Λ и входит в $< \Lambda >_A$. Таким образом, $< \Lambda >_A = A \cap \Theta$ и является линейным подпространством A.

Необходимость докажем методом от противного: предположим, что комбинаторная оболочка $< \Lambda >_A$ является линейным подпространством A и при этом $I_{\{1\}}(\Lambda) \neq \oslash$. Это значит, что есть такая позиция i, на которой во всех векторах из Λ стоит единица, а значит и во всех векторах из $< \Lambda >_A$. Отсюда получаем, что в $< \Lambda >_A$ нет нулевого вектора, что противоречит предположению, что $< \Lambda >_A$ является линейным подпространством. □

Теперь рассмотрим свойства комбинаторной зависимости на множестве столбцов $\mathcal{A}_a$.

Замечание. Зафиксируем порождающую матрицу $G^{(1)}$ кода Адамара и рассмотрим другую порождающую матрицу $G^{(2)}$ этого же кода. Тогда найдется такая обратимая матрица S, что $SG^{(1)} = G^{(2)}$. Нетрудно показать, что в этом случае найдется такая перестановочная матрица P, что $SG^{(1)} = G^{(1)}P$, т.е. $G^{(2)} = G^{(1)}P$ (см., например, [15], с. 228).

Лемма 3. *Мощность произвольного максимального комбинаторно-независимого подмножества в $\mathcal{A}_a$ больше a.*

Доказательство. Для того, чтобы комбинаторно-независимое множество было максимальным необходимо, чтобы его комбинаторная оболочка была равна $\mathcal{A}_a$, в противном случае можно взять вектор, который не попал в комбинаторную оболочку и получить другое комбинаторно-независимое множество, в которое вложено исходное, что противоречит определению максимального комбинаторно-независимого множества. Таким образом, $< \Lambda >_{\mathcal{A}_a} = \mathcal{A}_a$ и, следовательно, $< \Lambda >_{\mathcal{A}_a} = < \mathcal{A}_a >_{\mathcal{A}_a}$. По лемме 1 для любого непустого $D \subset \mathbb{F}_2^n$ имеет место равенство $I_D(\Lambda) = I_D(\mathcal{A}_a)$. Отсюда вытекает:

$$I_{\{0,1\}}(\Lambda) = \{2, \ldots, n\}, I_{\{0\}}(\Lambda) = \{1\}, I_{\{1\}}(\Lambda) = \oslash. \tag{5}$$

Рассмотрим все подмножества $\mathcal{A}_a$ мощности a и покажем, что ни одно из них не является максимальным комбинаторно-независимым подмножеством $\mathcal{A}_a$.

Сначала рассмотрим базис B пространства $\mathcal{A}_a$, описанный в разделе 1.2. Тогда

$$I_{\{0,1\}}(B) = \{2, \ldots, n-1\}, I_{\{0\}}(B) = \{1\}, I_{\{1\}}(B) = \{n\}.$$

В силу леммы 1 и равенств (5) $< B >_{\mathcal{A}_a} \neq < \Lambda >_{\mathcal{A}_a} = \mathcal{A}_a$, т.е. множество B не является максимальным комбинаторно-независимым подмножеством $\mathcal{A}_a$.

От базиса B можно перейти к любому другому базису B' с помощью последовательности элементарных преобразований, т.е. умножением слева на некоторую обратимую матрицу.

В силу замечания в результате перехода от B к B' фактически переставляются координаты векторов из B. После перестановки столбцов для любого непустого $D \subset \mathbb{F}_2^n$ получим $|I_D(B)| = |I_D(B')|$, а значит $I_{\{1\}}(B') \neq I_{\{1\}}(\Lambda) = \oslash$ и, по лемме 1, $< B' >_{\mathcal{A}_a} \neq \mathcal{A}_a$. Следовательно, никакой базис в $\mathcal{A}_a$ не является максимальным комбинаторно-независимым множеством.

Рассмотрим множество $C \subset \mathcal{A}_a$ и предположим, что $|C| = a$ и $\mathrm{rank}(C) = k < a$, т.е. в C имеется подмножество C' из k линейно независимых векторов, а остальные $a - k$ векторов являются их линейными комбинациями. Тогда $|I_{\{0\}}(C)| = 2^{a-k}$, т.к. имеется 2^{a-k} координат, на которых у векторов из C', а значит и у их линейных комбинаций, стоят только нули. Так как $k < a$, то $|I_{\{0\}}(C)| > 1$, и, следовательно, $< C >_{\mathcal{A}_a} \neq < \Lambda >_{\mathcal{A}_a} = \mathcal{A}_a$, а значит C не является максимальным комбинаторно-независимым подмножеством $\mathcal{A}_a$.

Таким образом, никакое подмножество $\mathcal{A}_a$ мощности a не является максимальным комбинаторно-независимым подмножеством. □

Лемма 4. *Во множестве столбцов $\mathcal{A}_a$ аддитивной матрицы Адамара рассмотрим множество $\Lambda = \{\lambda^{(i)}\}_{i=1}^{w}$.*

1. *Если Λ' – максимальное комбинаторно-независимое подмножество Λ, то $< \Lambda >_{\mathcal{A}_a} = < \Lambda' >_{\mathcal{A}_a}$.*
2. *$|< \Lambda >_{\mathcal{A}_a}| = 2^{\nu}$ для некоторого ν.*
3. *Пусть $w \leq a + 1$. Если Λ – комбинаторно-независимое множество, то $|< \Lambda >_{\mathcal{A}_a}| = 2^{w-1}$.*
4. *Если Λ' и Λ'' – различные максимальные комбинаторно-независимые подмножества Λ, то $|\Lambda'| = |\Lambda''|$.*

Доказательство.

1. Предположим, что это не так. Тогда $< \Lambda >_{\mathcal{A}_a} \supsetneq < \Lambda' >_{\mathcal{A}_a}$. Следовательно в $< \Lambda >_{\mathcal{A}_a}$ существует элемент a, который комбинаторно зависит от Λ, но не зависит от Λ', а значит существует такая позиция i, что $a_i \in \Lambda_i = \{\lambda_i | \lambda \in \Lambda\}$, но $a_i \notin \Lambda'_i = \{\lambda'_i | \lambda' \in \Lambda'\}$. Из максимальности Λ' вытекает, что все элементы из Λ комбинаторно-зависимы от Λ', и поэтому $\Lambda_i = \Lambda'_i$ для любого i. Из полученного противоречия вытекает $< \Lambda >_{\mathcal{A}_a} = < \Lambda' >_{\mathcal{A}_a}$.

2. Пусть $\Lambda' = \{\lambda'^{(i)} = \lambda^{(i)} - \lambda^{(1)}\}_{i=1}^{w}$. Тогда

$$I_{\{0,1\}}(\Lambda') = I_{\{0,1\}}(\Lambda),\ I_{\{0\}}(\Lambda') = I_{\{0\}}(\Lambda) \cup I_{\{1\}}(\Lambda),\ I_{\{1\}}(\Lambda') = \oslash. \tag{6}$$

   При этом $| < \Lambda >_{\mathcal{A}_a} | = | < \Lambda' >_{\mathcal{A}_a} |$. Действительно, $< \Lambda >_{\mathcal{A}_a}$ состоит из всех столбцов множества $\mathcal{A}_a$, у которых одновременно стоят нули в позициях $I_{\{0\}}(\Lambda)$ и единицы в позициях $I_{\{1\}}(\Lambda)$. С другой стороны, $< \Lambda' >_{\mathcal{A}_a}$ состоит из всех столбцов $\mathcal{A}_a$, у которых стоят нули в позициях $I_{\{0\}}(\Lambda) \cup I_{\{1\}}(\Lambda)$. Векторы из множества $< \Lambda' >_{\mathcal{A}_a}$ могут быть получены из множества векторов $< \Lambda >_{\mathcal{A}_a}$ вычитанием вектора $\lambda^{(i)}$. Таким образом, $| < \Lambda >_{\mathcal{A}_a} | = | < \Lambda' >_{\mathcal{A}_a} |$.

   В силу (6) по лемме 2 $< \Lambda' >_{\mathcal{A}_a}$ является линейным подпространством в $\mathcal{A}_a$, следовательно, мощность $< \Lambda >_{\mathcal{A}_a}$ равна степени двойки.

3. Докажем теперь, что $|< \Lambda >_{\mathcal{A}_a}| = 2^{w-1}$.

   При $w = 1$ утверждение верно: комбинаторная оболочка одноэлементного множества состоит из этого одного элемента, $2^{w-1} = 1$.

   Пусть $w > 1$. Рассмотрим множество $\Lambda' = \Lambda \backslash \{b\}$, где $b = \lambda^{(i)}$ для некоторого i. Множество Λ' очевидно комбинаторно-независимо, и в силу утверждения 2 $| < \Lambda' >_{\mathcal{A}_a} | = 2^{\nu'}$ для некоторого ν'. При этом элемент $b \in < \Lambda >_{\mathcal{A}_a}$, но $b \notin < \Lambda' >_{\mathcal{A}_a}$, т.к. множество Λ комбинаторно-независимо. Получаем, что

$$| < \Lambda >_{\mathcal{A}_a} | = 2^{\nu} > | < \Lambda' >_{\mathcal{A}_a} | = 2^{\nu'}, \tag{7}$$

   и, следовательно, $\nu > \nu'$.

   С другой стороны $< \Lambda >_{\mathcal{A}_a} \subseteq \mathcal{A}_a$, т.е. $|< \Lambda >_{\mathcal{A}_a}| \leq |\mathcal{A}_a| = n = 2^a$. При этом в силу (4) и утверждения 1 равенство достигается в случае, когда Λ является максимальным комбинаторно-независимым подмножеством $\mathcal{A}_a$.

   Предположим для начала, что $w \leq a + 1$. Построим из множества Λ цепочку множеств, $\{\Lambda^{(i)}\}_{i=1}^{a+1}$ по следующему правилу:

$$\Lambda^{(i)} = \begin{cases} \Lambda, & i = w \\ \Lambda^{(i+1)} \backslash \{b\}, & i < w, b \in \Lambda^{(i+1)} \\ \Lambda^{(i-1)} \cup \{c\}, & i > w, c \in \mathcal{A}_a \backslash < \Lambda^{(i-1)} >_{\mathcal{A}_a} \end{cases}.$$

   Для i от 0 до w такая цепочка, очевидно, может быть построена, а возможность построения цепочки для значений от $w + 1$ до $a + 1$ вытекает из леммы 3. Отметим, что $|\Lambda^{(i)}| = i$ и $\Lambda^{(i)} \subset \Lambda^{(i+1)}$.

Рассмотрим мощности комбинаторных оболочек множеств $\Lambda^{(i)}$:

$$1 = 2^0 = |< \Lambda^{(1)} >_{\mathcal{A}_a}| < |< \Lambda^{(2)} >_{\mathcal{A}_a}| < \cdots < |< \Lambda^{(a+1)} >_{\mathcal{A}_a}| \leq 2^a.$$

Согласно утверждению 2 мощность каждого из $< \Lambda^{(i)} >_{\mathcal{A}_a}$ равна степени двойки и $|< \Lambda^{(i)} >_{\mathcal{A}_a}| < |< \Lambda^{(i+1)} >_{\mathcal{A}_a}|$. Отсюда вытекает, что $|< \Lambda^{(i)} >_{\mathcal{A}_a}| = 2^{i-1}$, и для $i = w$ получаем доказываемое утверждение.

Покажем теперь, что случай $w > a + 1$ – невозможен. Построим аналогичную цепочку $\{\Lambda^{(i)}\}_{i=1}^{w}$

$$\Lambda^{(i)} = \begin{cases} \Lambda, & i = w \\ \Lambda^{(i+1)} \backslash \{b\}, & i < w, b \in \Lambda^{(i+1)} \end{cases}$$

и рассмотрим неравенства

$$1 = 2^0 = |< \Lambda^{(1)} >_{\mathcal{A}_a}| < |< \Lambda^{(2)} >_{\mathcal{A}_a}| < \cdots < |< \Lambda^{(w)} >_{\mathcal{A}_a}| \leq 2^a.$$

Здесь мы имеем $w > a + 1$ множеств, мощности которых различаются и равны степеням двойки, но только $a + 1$ степеней от 0 до a, отсюда получаем противоречие.

4. Для доказательства последнего утверждения предположим противное. Из утверждения 3 вытекает, что если Λ' и Λ'' имеют разные мощности, то и мощности их комбинаторных оболочек будут различными. Но из утверждения 1 получаем, что их комбинаторные оболочки должны быть равны комбинаторной оболочке множества Λ, т.е. иметь одинаковую мощность. Противоречие.

□

Лемма 4 позволяет ввести определение комбинаторного ранга $\mathrm{rank}_{\mathrm{comb}}(\Lambda)$ множества Λ как мощность любого его максимального комбинаторно-независимого подмножества и получить следующую теорему.

Теорема 2. *Рассмотрим множество столбцов аддитивной матрицы Адамара $\mathcal{A}_a$ и рассмотрим в нем произвольное множество Λ. Мощность комбинаторной оболочки Λ в множестве $\mathcal{A}_a$*

$$|< \Lambda >_{\mathcal{A}_a}| = 2^{\mathrm{rank}_{\mathrm{comb}}(\Lambda)-1}.$$

3. Вероятности успешного проведения коалиционных атак в системе распределения ключей в случае превышения предусмотренной мощности коалиции

Для оценки стойкости, построенной в разделе 1., системы распределения ключей вычислим вероятности успешного проведения коалиционной атаки в зависимости от мощности коалиции злоумышленников.

Будем считать, что коалиция злоумышленников формируется заранее, до раздачи пользователям идентификационных номеров и секретных ключей, а не подбирается специально для проведения атак на конкретные переписки. Это условие может быть достигнуто с помощью периодической смены идентификационных номеров.

Рассмотрим модель коалиционных атак. После того, как все участники системы получили идентификационные векторы – столбцы матрицы Адамара, коалиция злоумышленников может проверить, полный набор ключей каких пользователей системы они могут составить из объединения своих ключей. Для этого они могут построить множества $I_{\{0\}}(W)$ и $I_{\{1\}}(W)$, где W – множество идентификационных номеров–столбцов злоумышленников. С помощью этих множеств они могут построить комбинаторную оболочку $< W >_{\mathcal{A}_a}$, проверив идентификационные векторы каких пользователей имеют на всех позициях из $I_{\{0\}}(W)$ нули, а на всех позициях из $I_{\{1\}}(W)$ – единицы. Если идентификационный номер пользователя попадает в комбинаторную оболочку $< W >_{\mathcal{A}_a}$, то коалиция может вычислить ключ любой переписки с его участием.

Рассмотрим теперь вероятность того, что некоторая коалиция W мощности w может получить общий секретный ключ двух произвольных участников. Будем считать, что коалиция фиксированная, а сервер распределяет идентификационные номера случайным образом.

Прежде всего опишем пространства событий, необходимые для дальнейшего рассуждения. Введем Ω^S – пространство элементарных событий, каждое из которых является одним из возможных вариантов распределения сервером идентификационных номеров между пользователями системы, $|\Omega^S| = n!$. Будем считать, что все элементарные исходы равновероятны.

Введем также Ω^T – пространство элементарных событий, каждое из которых соответствует выбору произвольной пары пользователей T, на переписку которой коалиция хочет провести атаку, $|\Omega^T| = C_n^2$. Будем как и выше считать, что все элементарные исходы равновероятны.

Рассмотрим теперь пространство $\Omega = \Omega^S \times \Omega^T$ и посчитаем вероятность наступления события, состоящего в том, что коалиция W может получить общий секретный ключ пары T. Для любого элементарного исхода $\omega = (\omega^S, \omega^T) \in \Omega$ множество столбцов, соответствующих идентификационным номерам злоумышленников из W, будем обозначать $W(\omega^S) \subset \mathcal{A}_a$, а столбцы, соответствующие идентификационным номерам пользователей из $T(\omega^T)$, будем обозначать $T(\omega^T, \omega^S) \subset \mathcal{A}_a$. По теореме 1 для того, чтобы коалиция могла получить общий ключ T необходимо, чтобы коалиция получила все ключи одного из участников T, т.е. чтобы вектор одного из участников лежал в комбинаторной оболочке векторов злоумышленников из W. Таким образом, событие, состоящее в том, что коалиция W может получить общий ключ переписки произвольной пары T, можно записать так:

$$\Omega_{\text{comp}} = \{\omega \in \Omega | < W(\omega) >_{\mathcal{A}_a} \cap T(\omega) \neq \oslash\}.$$

Отметим, что это событие можно разбить на два события:

$$\Omega_{\text{self}} = \{\omega \in \Omega | W(\omega) \cap T \neq \oslash\},$$

которое наступает в том случае, когда один из пользователей из T входит в коалицию злоумышленников из W, и

$$\Omega_{\text{at}} = \{\omega \in \Omega | W(\omega) \cap T(\omega) = \oslash \wedge < W(\omega) >_{\mathcal{A}_a} \cap T(\omega) \neq \oslash\},$$

когда коалиция может получить общий ключ T только в результате проведения коалиционной атаки.

Теперь вычислим вероятности проведения успешных коалиционных атак. Для вычисления вероятности наступления интересующего нас события Ω_{at} введем множество вспомогательных гипотез $\{H(W,r)\}_{r=1}^{w}$, которые состоят в том, что множество W имеет комбинаторный ранг r, т.е.

$$H(W,r) = \{\omega \in \Omega | \ \text{rank}_{\text{comb}}(W(\omega)) = r\}.$$

Из-за случайного распределения столбцов матрицы Адамара, т.е. идентификационных номеров системы распределения ключей, вне зависимости от выбора коалиции W множество идентификационных номеров злоумышленников $W(\omega)$ будет пробегать все возможные подмножества $\mathcal{A}_a$ мощности w. Поэтому вероятность того, что коалиция W получит множество векторов комбинаторного ранга r, т.е. выполнится гипотеза $H(W,r)$, не зависит от W, а только от мощности w. Поэтому будем записывать эту гипотезу $H(w,r)$.

Вероятность наступления гипотезы $H(w,r)$ будем вычислять рекурсивно по w и r. Множество мощности w, имеющее комбинаторный ранг $r = \text{rank}_{\text{comb}}(W)$, может быть получено из множеств мощности $w-1$, которые разбиваются на два класса: множества с комбинаторным рангом $r-1$ и множества с комбинаторным рангом r. Для этого к множеству W'_{r-1} из первого класса нужно добавить вектор, комбинаторно-независимый от всех элементов этого множества, т.е. не входящий в его комбинаторную оболочку, а к множеству W'_r из второго класса – комбинаторно-зависимый, т.е. вектор, принадлежащий комбинаторной оболочке множества, но не принадлежащий самому множеству. Таким образом, если обозначить за $|H(w,r)|$ количество элементарных исходов благоприятствующих гипотезе $H(w,r)$, то получим следующую формулу:

$$|H(w,r)| = \frac{|H(w-1,r-1)| \cdot (|\mathcal{A}_a \backslash < W'_{r-1} >_{\mathcal{A}_a}|) + |H(w-1,r)| \cdot (|< W'_r >_{\mathcal{A}_a} \backslash W'_r|)}{w} =$$

$$= \frac{|H(w-1,r-1)| \cdot (2^a - 2^{r-2}) + |H(w-1,r)| \cdot (2^{r-1} - w + 1)}{w},$$

т.к. $|\mathcal{A}_a| = 2^a$, а в силу теоремы 2 $|< W >_{\mathcal{A}_a}| = 2^{\text{rank}_{\text{comb}}(W)-1}$, причем $\text{rank}_{\text{comb}}(W'_{r-1}) = r-1$, $\text{rank}_{\text{comb}}(W'_r) = r$.

Теорема 3. *Вероятность наступления события Ω_{at} может быть вычислена по формуле:*

$$p(\Omega_{at}) = \sum_{r=1}^{w} p(\Omega_{at} | H(w,r)) \cdot |H(w,r)| / C_n^w, \tag{8}$$

$$p(\Omega_{at} | H(w,r)) = \frac{C^1_{2^{r-1}-w} \cdot C^1_{n-2^{r-1}} + C^2_{2^{r-1}-w}}{C^2_n} \tag{9}$$

Доказательство. Гипотезы $H(w,r)$, $r = 1..w$ несовместны и в сумме образуют все вероятностное пространство Ω. Применив формулу полной вероятности

$$p(\Omega_{at}) = \sum_{r=1}^{w} p(\Omega_{at}|H(w,r)) \cdot p(H(w,r)),$$

и в силу того, что $p(H(w,r)) = |H(w,r)|/C_n^w$, получим (8). Формула (9), соответствующая вероятности того, что один или оба участника переписки попали в комбинаторную оболочку коалиции при условии наступления гипотезы $H(w,r)$, естественно вычисляется по формуле гипергеометрического распределения. □

Список литературы / References

[1] Шнайер Б., *Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си*, Триумф, 2002; [Schneier B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Inc., 1993.]

[2] Diffie W., Hellman M., "New Directions in Cryptography", *IEEE Transactions in Information Theory*, **22**:6 (1976), 644–654.

[3] Needham R. M. Schroeder M. D., "Using Encryption for Authentication in Large Networks of Computers", *Communications of the ACM*, **21**:12 (1978), 993–999.

[4] Matsumoto T., Imai I., "On the Key Predistribution System: A Practical Solution to the Key Distribution Problem", *CRYPTO '87 A Conference on the Theory and Applications of Cryptographic Techniques*, 1987, 185–193.

[5] Blom R., "An Optimal Class of Symmetric Key Generation Systems", *Workshop on the Theory and Applications of Cryptographic Techniques*, 1985, 335–338.

[6] Blundo C., Mattos L. A. F., Stinson D. R., "Trade-offs Between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution", *Annual International Cryptography Conference*, **1109** (1996), 387–400.

[7] Сидельников В. М., *Теория кодирования*, ФИЗМАТЛИТ, 2008; [Sidelnikov V. M., *Teoriya kodirovaniya*, FIZMATLIT, 2008, (in Russian).]

[8] Mitchell C. J., Piper F. C., "Key Storage in Secure Networks", *Discrete Applied Mathematics*, **21**:3 (1988), 215–228.

[9] Тарапников Ю. В., *Комбинаторные свойства дискретных структур и приложения к криптологии*, МЦНМО, 2011; [Tarannikov Yu. V., *Kombinatornye svoystva diskretnykh struktur i prilozheniya k kriptologii*, MTsNMO, 2011, (in Russian).]

[10] Stinson D. R., "On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption", *Designs, Codes and Cryptography*, **3**:12 (1997), 215–243.

[11] Stinson D. R., Trung T. V., "Some New Results on Key Distribution Patterns and Broadcast Encryption", *Designs, Codes and Cryptography*, **14** (1998), 261–279.

[12] Martin K M., "The Combinatorics of Cryptographic Key Establishment", *London Mathematical Society Lecture Note Series*, **346** (2007), 223–273.

[13] Деундяк В. М., Таран А. А., "О применении кодов Хэмминга в системе распределения ключей для конференций в многопользовательских системах связи", *Вестник ВГУ. Серия: Сист. анализ и информ. технологии*, **3** (2015), 43–50; [Deundyak V. M., Taran A. A., "O primenenii kodov Khemminga v sisteme raspredeleniya klyuchey dlya konferentsy v mnogopolzovatelskikh sistemakh svyazi", *Vestnik VGU. Seriya: Sist. analiz i inform. tekhnologii*, **3** (2015), 43–50, (in Russian).]

[14] Деундяк В. М., Таран А. А., "О вероятности проведения успешных атак на ключи конференций в полилинейных системах распределения ключей", *Известия вузов. Сев.-Кавк. Регион. Техн. Науки*, **1** (2018), 10–17; [Deundyak V. M., Taran A. A., "O veroyatnosti provedeniya uspeshnykh atak na klyuchi konferentsy v polilineynykh sistemakh raspredeleniya klyuchey", *Izvestiya vuzov. Sev.-Kavk. Region. Tekhn. Nauki*, **1** (2018), 10–17, (in Russian).]

[15] Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А., *Теория кодов, исправляющих ошибки*, Связь, 1979; [MacWilliams F. J., Sloane N. J. A., *The Theory of Error-Correcting Codes*, **16**, Elsevier, 1977.]

[16] Silverberg A., Staddon J., Walker J. L., "Applications of List Decoding to Tracing Traitors", *IEEE Transactions on Information Theory*, **49**:5 (2003), 1312–1318.

Deundyak V. M., Taran A. A., "Key Distribution System Based on Hadamard Designs", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 229–243.

DOI: 10.18255/1818-1015-2019-2-229-243

Abstract. The problem of key distribution in a community for providing secure communication between its participants is studied. To solve this problem, key predistribution systems can be used, in which each user receives some key information that can later be used to independently calculate required shared secret keys for conferences they participate in. Such key distribution systems can be based on different structures, such as error-correcting codes and combinatorial designs. The drawback of such systems is the possibility of collusive attacks, when traitors within the system can form a coalition and use their key information to try to calculate shared secret keys of other users. But the secrecy of keys is guaranteed by the system when the number of traitors in the coalition does not exceed a threshold defined by the system structure. In this paper, a key distribution system is based on combinatorial designs and, in particular, on Hadamard 3-design that guarantees the secrecy of communications in the presence of coalitions with less than three users. New notions of combinatorial span and combinatorial rank of a subset of Hadamard code that are required for the study of the resilience of the system to collusive attacks are introduced. The probability of successful collusive attack on an arbitrary conference against the cardinality of coalition is calculated for this system.

Keywords: key distribution systems, combinatorial designs, collusive attacks

On the authors:
Vladimir M. Deundyak, orcid.org/0000-0001-8258-2419, PhD,
Southern Federal University,
105/42 Bolshaya Sadovaya str., Rostov-on-Don 344006, Russia,
FGNU NII "Specvuzavtomatika",
51 Gazetniy lane, Rostov-on-Don 344002, Russia, e-mail: vl.deundyak@gmail.com

Alexey A. Taran, orcid.org/0000-0002-1357-9360, graduate student,
Southern Federal University,
105/42 Bolshaya Sadovaya str., Rostov-on-Don 344006, Russia, e-mail: fraktal-at@yandex.ru

Моделирование и анализ информационных систем. Т. 26, №2 (2019), с. 244–255
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 244–255

Theory of computing

DOI: 10.18255/1818-1015-2019-2-244-255
УДК 51-37

К вопросу использования «полезных» задач для обеспечения работой блокчейн систем

Мурин Д. М., Князев В. Н.

Поступила в редакцию 13 марта 2019
После доработки 17 мая 2019
Принята к публикации 20 мая 2019

Аннотация. Статья является продолжением работы о возможных подходах к решению задачи «Useful Proof-of-work for blockchains». Мы предлагаем некоторые альтернативные направления поиска полезных задач для обеспечения работой, основанные на том, что процесс решения хеш-головоломки близок к многократному независимому повторению следующего эксперимента: пусть задано достаточно большое по мощности множество (например, состоящее из 2^n элементов, для достаточно большого n), только незначительная часть элементов которого обладает определенным свойством. Эксперимент состоит в равномерном выборе элемента из этого множества с последующей проверкой наличия у него указанного свойства. Таким образом, процесс решения хеш-головоломки может быть заменен, например, поиском редких астрономических объектов или поиском позиций игры Го, удовлетворяющих определенным условиям. Кроме того, мы описываем возможную атаку на блокчейн-систему, в которой алгоритм генерации индивидуальных представителей задач для обеспечения работой заменен алгоритмом выбора индивидуальных представителей из имеющейся базы данных, со стороны недобросовестных поставщиков индивидуальных представителей задач, в случае их публичного сбора, и обсуждаем некоторые способы защиты от этой атаки.

Ключевые слова: доказательство работой, блокчейн, алгоритм

Для цитирования: Мурин Д. М., Князев В. Н., "К вопросу использования «полезных» задач для обеспечения работой блокчейн систем", *Моделирование и анализ информационных систем*, **26**:2 (2019), 244–255.

Об авторах:
Мурин Дмитрий Михайлович, orcid.org/0000-0002-8068-0784, канд. физ.-мат. наук, доцент кафедры КБиММОИ,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: nirum87@mail.ru

Князев Владимир Николаевич, orcid.org/0000-0003-4967-0825, ассистент кафедры КБиММОИ
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: darknyaz@yandex.ru

1. Модель классического блокчейна

Напомним некоторые определения.

Определение 1. *Система* ***Proof-of-Work (PoW)*** *— это 3 алгоритма:*

1. `Gen`(I) — *вероятностный полиномиальный по времени алгоритм, получающий на вход двоичное слово длины* $n - I$ *и выводящий индивидуального представителя* TI *некоторой задачи.*

2. `Solve`(TI) — *алгоритм, решающий индивидуального представителя* TI *и выводящий (в случае завершения работы) его решение* S*.*

3. `Verify`(TI, S) — *вероятностный полиномиальный по времени алгоритм, проверяющий решение* S *индивидуального представителя задачи* TI*.*

Алгоритм `Solve` *должен иметь известную сложность в среднем* $t(n)$*, причем любые алгоритмы со сложностью в среднем, асимптотически меньшей, чем* $t(n)$*, не должны выдавать истинные решения* TI*, а приближенные решения, выдаваемые такими алгоритмами, должны с большой вероятностью отвергаться алгоритм* `Verify`*.*

Системы PoW, предназначенные для использования в блокчейн технологиях, имеют несколько отличающееся определение. Мы будем называть такие системы **Proof-of-Work-for-Blockchain** (**PoWB**).

Определение 2. *Система PoW называется системой* ***PoWB****, если*

1. *Алгоритм* `Gen` *получает на вход двоичное слово длины* $n - I$ *и натуральное число* $- C$ *и возвращает индивидуального представителя задачи* TI*, как и в системе PoW.*

2. *Сложность в среднем алгоритма* `Solve` *зависит от входа* C*. Изменяя* C *можно управлять сложностью алгоритма* `Solve`*.*

Определение 3. *Система PoW называется системой* ***Useful Proof-of-Work (UPoW)****, если алгоритм* `Gen` *генерирует индивидуальных представителей полезных за пределами блокчейна задач.*

Аналогичным образом вводится определение для систем **Useful Proof-of-Work-for-Blockchain** (**UPoWB**).

Определение 4. *Блокчейн, в котором система PoW является системой UPoWB, мы будем называть* ***UPoWB-блокчейном****.*

Определение 5. ***Блокчейн*** *(от англ. blockchain, буквально — «цепочка блоков») — растущий список записей, называемых* ***блоками****, которые криптографически связаны друг с другом.*

Под классическим блокчейном мы, в первую очередь, понимаем блокчейн системы Bitcoin, описанный в работе [6], упрощенная схема которого представлена на рис. 1. Классический блокчейн состоит из следующих компонентов:

1. **Система PoWB**. Система PoWB обеспечивает формирование блоков с определенной периодичностью.

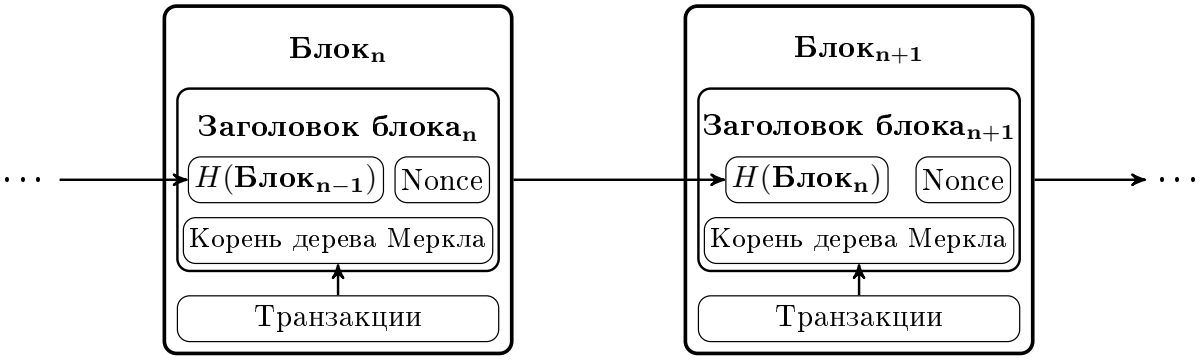


Рис. 1: Классический блокчейн
Fig. 1: Classical blockchain

2. **Транзакция**. В системе Bitcoin транзакции хранят информацию о переводах денежных средств. В общем случае это некоторые данные, чья целостность и аутентичность обеспечиваются с помощью электронной подписи. Транзакции делятся на два вида: *неподтвержденные* и *подтвержденные*. При создании транзакция считается неподтвержденной. Для изменения вида транзакция должна быть связана с блоком, а блок, в свою очередь, должен войти в «победившую» цепочку блоков.

3. **Блок**. Состоит из заголовка блока и списка, вошедших в блок транзакций. Служит для обеспеченного вычислительной работой подтверждения транзакций путем их криптографического связывания с блоком с помощью алгоритма `Solve` системы PoWB. Каждый блок криптографически связан с предыдущим блоком, таким образом, блоки образуют связанный список — *блокчейн.*

4. **Блокчейн-сеть**. Представляет собой распределенную вычислительную систему, вычислительные узлы которой практически все время решают индивидуальных представителей TI задач системы PoWB для подтверждения транзакций и периодически обмениваются между собой информацией в целях синхронизации и актуализации данных о сгенерированных блоках.

Отметим, что блокчейн-сеть обычно разбивается на вычислительные **кластеры** (**пулы**), в каждом из которых решается один и тот же индивидуальный представитель задачи. Пусть блокчейн-сеть обладает зависящей от времени вычислительной мощностью $M(t)$ и в момент времени t существует $K(t)$ кластеров, каждый их которых обладает идентификатором — числом из множества $\{1, \ldots, K(t)\}$, и i-й кластер обладает вычислительной мощностью $M_i(t)$. Вычислительная мощность блокчейн-сети $M(t) = \sum_{i=1}^{K(t)} M_i(t)$, но вычислительная мощность, направленная на решение одного индивидуального представителя TI, не превосходит $\max_{1 \le i \le K(t)} M_i(t)$.

Также в каждом кластере (а возможно и на каждом вычислительном узле) хранится копия цепочки блоков, поэтому по сути она является распределенной базой данных. В каждом вычислительном кластере в момент времени t может формироваться собственный блок для хранимой цепочки блоков, но при получении в процессе обмена с другими кластерами информации о более длинной цепочке блоков, кластер обязан переключиться на работу с ней, отказавшись от результатов своей

предыдущей работы. Одновременно может существовать несколько различных цепочек блоков, но длина каждого альтернативного ответвления от основной цепочки не может превышать некоторой величины $Q(t)$. Времени, за которое формируются $Q(t)$ блоков, должно быть достаточно для оповещения всех кластеров о наличии более длинной цепочки (для блокчейн-системы Bitcoin на момент написания статьи $Q(t) = 6$). Таким образом, одного блока для подтверждения транзакции работой недостаточно. Транзакция считается подтвержденной в случае, если она включена в блок, за которым была сформирована цепочка из $Q(t)$ блоков. Это сделано для уменьшения вероятности **атаки двойного расходования** [6].

2. О некоторых альтернативах хеш-головоломкам

В качестве задач для обеспечения работой в классическом блокчейне используются задачи поиска двоичного слова, хеш-значение от которого обладает определенными свойствами хеш-головоломки. В качестве примера мы рассмотрим вариант хеш-головоломок, для которого хеш-значение должно иметь начало из не менее l_1 нулей (в более «эластичном» варианте хеш-значение должно быть ассоциировано с натуральным числом меньшим, чем l_2). l_1 и l_2 являются управляющими параметрами блокчейн-системы, отвечающими за сложность решаемых индивидуальных представителей хеш-головоломок.

Использование хеш-головоломок в блокчейн-системах позволяет решить две главные задачи: криптографически привязать транзакции к блоку и обеспечить транзакции вычислительной работой. Обратим внимание, что эти задачи могут решаться разными способами (не обязательно искать методы решения обеих задач сразу, для каждой задачи может быть предложен свой вариант решения). Основным недостатком хеш-головоломок является их бесполезность за пределами блокчейн-систем.

Рассмотрим более подробно процесс решения хеш-головоломки.

Пусть $H : \{0,1\}^n \to \{0,1\}^m$ является качественной криптографической хеш-функцией. Выходное распределение, порождаемое функцией H на равномерном входе — $H(U_n)$, где U_n — случайная величина, равномерно распределенная на $\{0,1\}^n$, должно быть вычислительно неразличимо с равномерным распределением на множестве $\{0,1\}^m$. При решении хеш-головоломки (за неимением хороших алгоритмов решения, существенно отличающихся от перебора) необходимо многократно выбирать аргумент хеш-функции x и проверять, удовлетворяет ли хеш-значение $y = H(x)$ условию головоломки. Этот процесс близок к многократному повторению эксперимента с равномерным выбором $y \leftarrow U_m$ и последующей проверкой того, удовлетворяет ли этот элемент условию головоломки. При этом множество $\{0,1\}^m$ делится на 2 подмножества. Элементы первого подмножества удовлетворяют условиям головоломки, а элементы второго — нет (например, все двоичные слова длины m делятся на те, которые имеют начало из l_1 нулей, и те, которые не имеют).

Следовательно, у каждого эксперимента есть два возможных исхода и случайная величина X, которая принимает значение 1 с вероятностью p в случае успешного завершения эксперимента и 0 — в противном случае (с вероятностью $q = 1-p$), имеет распределение Бернулли. Определим случайную величину Y, которая принимает значение, равное номеру первого «успеха». Эта случайная величина имеет геомет-

рическое распределение: $Pr[Y = n] = q^{n-1}p$ и математическое ожидание $E[Y] = \frac{1}{p}$. Таким образом, в среднем нам необходимо провести $\frac{1}{p}$ экспериментов до получения первого успеха. Тогда, если мы случайно и равномерно выбираем слово длины n, то вероятность выбрать слово, имеющее начало из l_1 нулей, составляет $p = \frac{1}{2^{l_1}}$, и в среднем первое такое слово мы выберем через 2^{l_1} экспериментов.

По данным сайта www.blockchain.com, для блокчейн системы Bitcoin число проводимых экспериментов при решении хеш-головоломок достигало в 2017 году $3,6 * 10^{22} \simeq 2^{76}$ [2]. При этом порядка 2^{73} экспериментов могли быть проведены на одном кластере [3].

Предположим, что мы исследуем некоторые объекты или физические процессы, причем интерес для исследования представляют чрезвычайно редкие события (например, встречающиеся с вероятностью $p = \frac{1}{2^{l_1}}$). Тогда любое вновь обнаруженное событие могло бы выступать в качестве обеспечения работой для некоторой блокчейн системы. Например, для блокчейн системы Bitcoin такими объектами могли бы служить астрономические объекты, обладающие особыми, редкими свойствами (число звезд в наблюдаемых галактиках по оценкам составляет $\sim 10^{23}$, и обнаружение объектов, обладающих редкими свойствами, например, при анализе снимков, сделанных космическими телескопами, могло бы обеспечивать транзакции вычислительной работой), объекты возникающие при изучении элементарных частиц (редкие события, выявляемые при анализе треков частиц, получаемых с помощью адронного коллайдера), удовлетворяющие особым условиям шахматные партии (оценка их числа $\sim 10^{120}$ [4]) или позиции (оценка их числа $\sim 10^{43}$ [4]), позиции игры Го (оценка их числа для игрового поля размером 19 на 19 клеток $\sim 10^{171}$ [5]) и другие.

При этом алгоритм `Gen` должен указывать на некоторый массив данных, в котором должен осуществляться поиск редкого события или явления (например, на конкретный снимок или подмножество снимков, сделанных космическими телескопами, или множество шахматных партий, начинающегося с определенной последовательности ходов, и так далее). Алгоритм `Solve` производит перебор объектов из массива данных. Детерминированный алгоритм `Verify` проверяет, что обнаруженный объект действительно удовлетворяет необходимым требованиям.

Повторное использование одного и того же объекта можно исключить, осуществляя поиск по ранее обнаруженным объектам, поскольку подтверждение работой потребует публикации данных об обнаруженном объекте.

3. Некоторые подходы к построению UPoWB-блокчейна

В этом разделе мы опишем возможную атаку на блокчейн систему, предложенную в [1], со стороны недобросовестных поставщиков индивидуальных представителей задач в случае их публичного сбора, и обсудим некоторые способы защиты от этой атаки.

3.1. Разработанная модель UPoWB-блокчейна

Наша идея построения системы UPoWB основана на отказе от алгоритма «генерации» `Gen` индивидуального представителя задачи системы UPoWB. Вместо этого мы предполагаем, что имеется база данных индивидуальных представителей полезных задач (например, полученных в ходе практической деятельности). В базе данных в процессе предобработки необходимо выбрать таких индивидуальных представителей, решение которых можно принять в качестве обеспечения работой. То есть индивидуальных представителей, решение которых должно потребовать существенных вычислительных ресурсов. Мы считаем, что некоторый алгоритм `Solve` решения индивидуальных представителей является общеизвестным (например, для задач из класса NP в качестве такого алгоритма может выступать SAT-решатель), поэтому все узлы блокчейн-сети имеют возможность решать индивидуальных представителей. Также мы считаем, что решение индивидуального представителя задачи можно относительно быстро проверить с помощью алгоритма `Verify` (что, естественно, выполнено для задач из класса NP).

Компоненты разработанного блокчейна:

1–4. Совпадают с компонентами 1–4 классического блокчейна (см. раздел 1.), за исключением того, что в блок или дерево Меркла могут включаться требующие решения индивидуальные представители полезных задач.

5. **База данных индивидуальных представителей полезных задач**. В базе данных содержатся индивидуальные представители полезных задач требующие решения. Базу данных можно считать открытым ресурсом, находящимся под управлением третьей доверенной стороны — **администратора базы данных**. Администратора базы данных можно рассматривать как публичный сервис. Перед открытой публикацией индивидуальных представителей проводится их предварительный анализ (предобработка), от которой требуется исключение из рассмотрения легких индивидуальных представителей и, по возможности, выбор представителей, которые могут быть решены за определенный интервал времени системой блокчейн.

3.2. Атака на систему со стороны недобросовестных поставщиков задач

В работе [1] предложены некоторые подходы к использованию NP-полных задач в качестве задач для обеспечения работой. В одном из подходов предлагается формировать базу данных индивидуальных представителей задач для обеспечения работой из «внешних» источников. При этом в указанной работе неявно предполагалась либо добросовестность поставщиков (источников) задач, либо достаточная сила алгоритма управления сложностью задач, позволяющая исключать из рассмотрения задачи недобросовестных поставщиков. Если ни одно из этих условий не обеспечивается, то следует учитывать описанную далее атаку. Будем считать, что индивидуальные представители независимо и равномерно выбираются из множества доступных для решения индивидуальных представителей, находящихся в

базе данных. Выбор может быть основан на информации из текущего блока. Будем считать, что он зависит от выбранных транзакций.

Предположим, что в качестве источника задач может выступать злоумышленник. Тогда для обеспечения работой злоумышленник может выбирать индивидуальных представителей задач, которые просты для решения, маскировать их под индивидуальных представителей общего положения и предлагать их для включения в базу данных. То есть злоумышленник может пытаться предлагать для включения в базу данных индивидуальных представителей с известными ему секретами. Например, злоумышленник может выбрать индивидуального представителя задачи о рюкзаке, имеющего сверхрастущий вектор или вектор с малой плотностью, замаскировать его с помощью сильного модульного умножения (или другого преобразования) и предложить для включения в базу данных.

При этом злоумышленник, с нашей точки зрения, может преследовать две цели:

1. *Тривиальная.* Заключается в получении выгоды от формирования блока. При выборе для обеспечения работой индивидуального представителя, для которого злоумышленнику известен секрет, он (злоумышленник) получает преимущество в скорости формирования блока.

2. *Основная.* Получение выгоды от быстрого формирования альтернативной цепочки «тяжелых» блоков. Если доля индивидуальных представителей с секретами, известными злоумышленнику, в базе данных достаточно велика, то это обеспечивает высокую вероятность выбора выгодных злоумышленнику индивидуальных представителей для обеспечения работой, и злоумышленник получает возможность быстрого формирования достаточно длинной альтернативной цепочки «тяжелых» блоков (понятие «тяжелый» блок введено в работе [1]).

Рассмотрим случайную величину X', которая принимает значение 1 с вероятностью p', если выбран индивидуальный представитель задачи с секретом, известным злоумышленнику, и 0 — в противном случае (с вероятностью $q' = 1 - p'$). Вновь случайная величина X' имеет распределение Бернулли. Соответственно, случайная величина Y', которая принимает значение равное номеру первого «успеха» (то есть выбору индивидуального представителя с секретом, известным злоумышленнику), имеет геометрическое распределение. Математическое ожидание случайной величины $E[Y'] = \frac{1}{p'}$. Если $E[Y'] << E[Y]$, то злоумышленник следующим образом может быстро сформировать цепочку «тяжелых» блоков.

Злоумышленник случайным образом выбирает неподтвержденные транзакции для привязки к блоку, если выбор оказывается неудачным (то есть для полученной выборки транзакций ассоциированный индивидуальный представитель не обладает секретом, известным злоумышленнику), то злоумышленник выбирает новое множество транзакций. Поскольку среднее число экспериментов, необходимых для выбора индивидуального представителя с секретом $E[Y']$, значительно меньше, чем среднее число экспериментов, необходимых для решения хеш-головоломки $E[Y]$, то злоумышленник получает существенное преимущество при формировании цепочки «тяжелых» блоков.

Пример 1. *Пусть мы используем для формирования «легких» блоков хеш-головоломки с условием, что хеш-значение должно иметь начало из l_1 нулей. Предположим, что злоумышленник может обеспечить ситуацию, при которой из всего множества доступных для решения блокчейн-сети индивидуальных представи-*

телей $\frac{1}{poly(l_1)}$*-доля имеет известные ему секреты (*$poly(l_1)$ *— некоторый полином от* l_1*). Тогда «честному» пользователю для формирования одного «легкого» блока в среднем требуется* 2^{l_1} *экспериментов, трех «легких» блоков, эквивалентных одному «тяжелому» блоку (в соответствии с [1]),* $— 3 \cdot 2^{l_1}$*, а злоумышленнику для формирования одного «тяжелого» блока требуется* $poly(l_1) + 2^{l_1}$ *экспериментов.*

Для достижения обеих целей (особенно второй цели) злоумышленнику необходимо добиться, чтобы доля принятых в базу данных индивидуальных представителей с известными ему секретами была очень велика. Если мы не можем различать индивидуальных представителей с секретами (или, возможно, более широкие множества индивидуальных представителей, подмножествами которых являются индивидуальные представители с секретами) от задач общего положения на этапе управления сложностью задач (в противном случае, при обнаружении они подлежат исключению из рассмотрения), то необходимо предложить иные способы противодействия рассмотренной атаке.

3.3. Некоторые способы защиты от описанной атаки

Описанная в предыдущем разделе атака имеет мотив в виде увеличенного вознаграждения за «тяжелый» блок и возможна благодаря тому, что злоумышленник, зная решение индивидуального представителя с известным ему секретом, может *единолично* сформировать «тяжелый» блок (как описано в [1]).

Одним из возможных способов защиты от описанной атаки может являться маскирование полученной от поставщика задачи администратором базы данных индивидуальных представителей полезных задач. Например, полученный от поставщика индивидуальный представитель полезной задачи может быть полиномиально сведен в задачу выполнимости булевой формулы, в полученной формуле осуществлена случайная замена и случайная перестановка переменных, возможно, с раскрытием скобок или вынесением отдельных элементов за скобки. Дополнительно можно свести полученного индивидуального представителя задачи выполнимости в иную (случайно выбранную) NP-полную задачу с последующим сведением в выполнимость. Описанный способ может существенно затруднить злоумышленнику поиск «своих» индивидуальных представителей. Однако он обладает некоторыми недостатками. Во-первых, мы не можем гарантировать, что не существует индивидуальных представителей, сохраняющих определенную структуру после указанных преобразований, позволяющую эффективно установить связь с исходным индивидуальным представителем. Во-вторых, указанные преобразования могут приводит к увеличению размерности индивидуального представителя.

Делая бессмысленной основную цель злоумышленника, можно потребовать, чтобы «тяжелые» блоки формировались редко, например, не чаще, чем один на $Q(t)+1$ блоков, где $Q(t)$ такое же, как в разделе 1. В этом случае не только злоумышленник, но и все кластеры (узлы) блокчейн-сети лишаются возможности строить цепочки «тяжелых» блоков. При этом злоумышленник сохраняет возможность получить выгоду от формирования одного «тяжелого» блока. И введение платы за решение задачи (от поставщика индивидуального представителя — решившему задачу) не исправляет ситуации, поскольку злоумышленник в этом случае заплатит сам себе.

Мы видим основным способом защиты от описанной в предыдущем разделе атаки изменение схемы мотивации создания «тяжелого» блока. В [1] предлагалось оставить в блокчейн системе как «легкие», так и «тяжелые» блоки. При этом «легкие» блоки использовались в качестве своеобразного метронома. В измененной схеме мы потребуем, чтобы каждый блок нес полезную нагрузку, то есть был «тяжелым». Соответственно за формирование каждого блока устанавливается фиксированная плата. Кроме этого, мы разъединим процесс решения индивидуального представителя полезной задачи и процесс формирования блока. А именно, мы потребуем, чтобы решение индивидуального представителя из базы данных публиковалось в базе данных вместе с подписью администратора. С точки зрения компонентов блокчейн потребуются небольшие изменения: в блоке должно появиться поле для подписи администратора базы данных, а в самой базе данных должны будут публиковаться не только индивидуальные представители, но и их решения, подписанные электронной подписью администратора.

Обозначим через `Info`(**Block**) информацию, которой будет достаточно для формирования блока (например, `Info`(**Block**) может состоять из указателя на предыдущий блок, корня дерева Меркла и списка транзакций). Отметим, что мы также используем `Info`(**Block**) для выбора индивидуального представителя из базы данных (например, можно использовать хеш-значение от `Info`(**Block**) или его части в качестве указателя на индивидуального представителя в базе данных). Приведем более формальный алгоритм формирования «тяжелого» блока в модифицированном блокчейне.

Решение индивидуального представителя:

1. Выбрать с помощью `Info`(**Block**) индивидуального представителя из базы данных (например, способом, описанным в [1]).

2. Решить индивидуального представителя полезной задачи и направить решение вместе с информацией о решенном индивидуальном представителе и формируемом блоке `Info`(**Block**) в базу данных.

3. После публикации решения в базе данных индивидуальный представитель считается решенным.

Администратор базы данных, получая предлагаемые решения, может подписывать их и публиковать, оставляя проверку правильности решений кластерам блокчейн-сети. Также администратора можно наделить функцией проверки правильности решения. В этом случае администратор проверяет правильность выбора индивидуального представителя для формируемого блока и правильность самого решения. Если решение верно, то подписывает его и публикует.

Формирование блока:

1. Выбрать опубликованное в базе данных решение.

2. Путем решения хеш-головоломки сформировать блок, в котором содержится информация о решенном индивидуальном представителе, его решение и подпись администратора базы данных.

Таким образом, любой кластер может закончить формирование «тяжелого» блока, решив стандартную хеш-головоломку для опубликованного в базе данных блока, связанного с решенным индивидуальным представителем. Так как подпись администратора трудно спрогнозировать заранее, злоумышленник не будет обладать преимуществом в формировании «тяжелого» блока. Рис. 2 иллюстрирует наше решение.

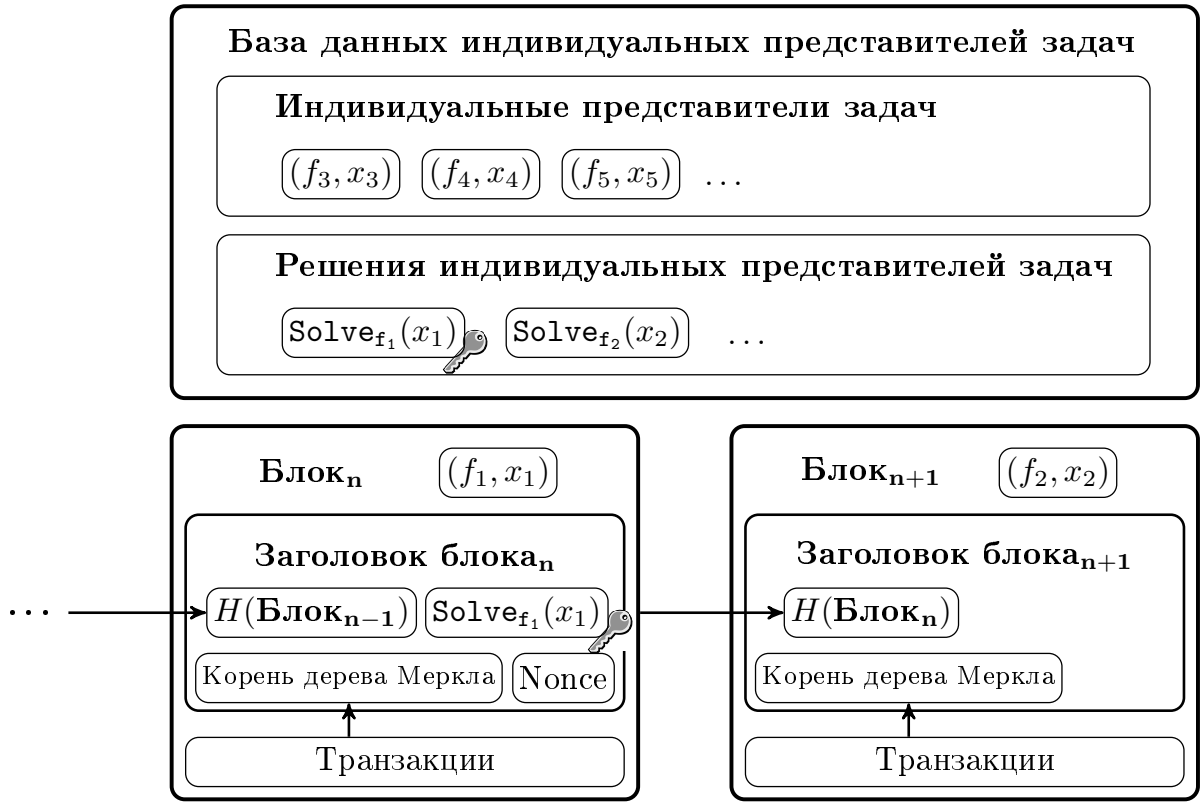


Рис. 2: Модель разработанного UPoWB-блокчейна. Рисунок иллюстрирует невозможность формирования $n + 1$ блока, так как ассоциированная с ним задача еще не прошла проверку администратором базы данных индивидуальных представителей полезных задач

Fig. 2: UPoWB-blockchain model. The picture illustrates the impossibility of forming $n + 1$ block since the associated task did not pass the check of individual representatives of useful tasks database administrator yet

В данном случае можно лишить смысла достижение тривиальной цели злоумышленника путем установления обязательной оплаты решения индивидуального представителя его поставщиком. Чтобы избежать выплаты вознаграждения злоумышленником себе самому, часть вознаграждения можно адресовать администратору базы данных. Таким образом, после опубликования решения кластер, предоставивший верное решение, получает за это награду, определенную поставщиком для соответствующего индивидуального представителя.

Отметим, что в модифицированном блокчейне в качестве полезной нагрузки могут выступать задачи любой сложности. Однако в одном интервале времени необхо-

димо обеспечить доступность для решения индивидуальных представителей, имеющих приблизительно одинаковую сложность.

4. Заключение. О «мгновенной» сложности задач

В заключение мы хотим отметить, что технология блокчейн позволяет ставить вопрос о «мгновенной» сложности решаемых задач. Фактически, для обеспечения работой нам необходим индивидуальный представитель, решение которого мы готовы принять в качестве такого обеспечения в текущем временном интервале. Вне этого интервала могут поменяться как внутренние факторы, например, мощность блокчейн-сети, так и внешние факторы, например, будет разработан новый существенно более быстрый алгоритм для решения индивидуальных представителей определенного вида. С точки зрения классической теории сложности каждый индивидуальный представитель решается за время $O(1)$. С точки зрения «мгновенной» сложности нам необходим более точный анализ числа шагов алгоритмов, решающих индивидуальных представителей задач.

Список литературы / References

[1] Дурнев В. Г., Мурин Д. М., Соколов В. А., Чалый Д. Ю., “О некоторых подходах к решению задачи «Useful Proof-of-work for blockchains»”, *Моделирование и анализ информационных систем*, **25**:4 (2018), 402–410; [Durnev V. G., Murin D. M., Sokolov V. A., Chalyy D. Ju., “On Some Approaches to the Solution of the Problem ”Useful Proof-of-work for Blockchains””, *Modeling and Analysis of Information Systems*, **25**:4 (2018), 402–410, (in Russian).]

[2] “Hash Rate. The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing.”, 2018, https://www.blockchain.com/ru/charts/hash-rate.

[3] “Распределение количества перебора хешей.”, 2018, https://www.blockchain.com/ru/pools; [“Hashrate quantity distribution.”, 2018, https://www.blockchain.com/ru/pools, (in Russian).]

[4] Shannon C., “Programming a Computer for Playing Chess”, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **41**:314 (1950), 256–275.

[5] “Number of legal Go positions.”, 2016, https://tromp.github.io/go/legal.html.

[6] Nakamoto S., “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008, https://bitcoin.org/bitcoin.pdf.

[7] Marshall Ball, Alon Rosen, Manuel Sabin, Prashant Nalini Vasudevan, “Proofs of Useful Work”, 2017, https://eprint.iacr.org/2017/203.pdf.

[8] “Problem 11. Useful Proof-of-work for blockchains”, 2017, https://nsucrypto.nsu.ru/archive/2017/round/2/section/0/task/11/.

Murin D. M., Knyazev V. N., "On the Issue of Using “Useful” Tasks for Proof of Works in Blockchain", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 244–255.

DOI: 10.18255/1818-1015-2019-2-244-255

Abstract. This paper is a logical continuation of the paper about possible approaches to solving the "Useful Proof-of-work for blockchains" problem. We suggest some alternative ways for searching useful tasks for Proof-of-work systems. These ways are based on the process of the multiple and independent repetition of a simple experiment. The experiment is to chose an element independently and uniformly from a quite large set and then to check if the chosen element has a specific rare property. In the classic blockchain of Bitcoin this experiment is a so-called hash-puzzle. In these terms the process of solving a hash-puzzle may be replaced by searching rare astronomical objects or Go positions with specific conditions. Moreover, we describe a possible attack on the blockchain systems in which the task instance generation algorithm is replaced by the algorithm of selecting the task instance from the existing database with public access for publication of task instances and discuss the way of protection.

Keywords: proof-of-work, blockchain, algorithm

On the authors:
Dmitry M. Murin, orcid.org/0000-0002-8068-0784, PhD, Docent,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: nirum87@mail.ru

Vladimir N. Knyazev, orcid.org/0000-0003-4967-0825, Assistant,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: darknyaz@yandex.ru

Моделирование и анализ информационных систем. Т. 26, №2 (2019), с. 256–266
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 256–266

DOI: 10.18255/1818-1015-2019-2-256-266

УДК 517.9

Анализ возможностей практического использования моделей решеточных газов

Бобков С. П., Чернявская А. С., Шергин В. В.

Поступила в редакцию 23 января 2019
После доработки 14 мая 2019
Принята к публикации 16 мая 2019

Аннотация. В последние годы для математического моделирования физико-химических процессов стали широко применяться дискретные подходы. Среди них исследователи выделяют методы, основанные на использовании клеточных автоматов. Привлекательность данных математических объектов обоснована прежде всего тем, что во многих случаях они существенно упрощают процедуры моделирования по сравнению с традиционными методами. В частности, при использовании моделей в виде систем дифференциальных уравнений с частными производными для анализа переноса субстанции, трудности возникают в случаях протекания процессов в неоднородных средах. Кроме того, в ряде случаев довольно проблематично осуществить корректную постановку граничных условий, если объект исследования имеет границы сложной формы. Также трудно использовать классические уравнения математической физики в условиях, когда невозможно игнорировать влиянии стохастических эффектов на протекание процесса. Дискретные подходы в значительной мере свободны от указанных недостатков. Рассматриваемые в статье модели решеточных газов являются одной из разновидностей клеточных автоматов. Несмотря на то, что первые работы по использованию решеточных моделей газов появились около сорока лет назад, они до настоящего времени не получили широкого распространения в среде исследователей естественнонаучных процессов. Тем не менее имеется много доказательств того, что решеточные газы достаточно успешно описывают целый ряд гидродинамических явлений, а полученные результаты не противоречат общепринятым взглядам на физическую природу процессов движения сплошных сред. Несмотря на появление значительного количества разновидностей моделей решеточных газов, при их использовании часто возникают вопросы, касающиеся режимов течения, при которых использование дискретных моделей будет корректным. Вторая проблема, обычно возникающая перед исследователями, использующими решеточные модели, – это масштабный переход от модельных дискретных параметров к общепринятым макроскопическим характеристикам течений. Здесь, прежде всего, имеются в виду такие физические величины, как скорость потока, вязкость и плотность среды и пр. Ситуация осложняется тем обстоятельством, что указанные параметры в решеточной модели являются безразмерными, а соответствующие реальные макроскопические показатели имеют размерность. В данной статье делается попытка предложить методику масштабного перехода, а также указать области практического использования некоторых моделей решеточных газов.

Ключевые слова: дискретный подход, решеточный газ, HPP, FHP и LBM модели

Для цитирования: Бобков С. П., Чернявская А. С., Шергин В. В., "Возможности использования моделей решеточных газов", *Моделирование и анализ информационных систем*, **26**:2 (2019), 256–266.

Об авторах:
Бобков Сергей Петрович, orcid.org/0000-0001-7315-1625, докт. техн. наук, профессор,
Ивановский государственный химико-технологический университет

Шереметевский пр., 7, г. Иваново, 153000 Россия, e-mail: bsp@isuct.ru

Чернявская Анастасия Сергеевна, orcid.org/0000-0001-9717-9665, ассистент,
Ивановский государственный химико-технологический университет
Шереметевский пр., 7, г. Иваново, 153000 Россия, e-mail: gbonnefee@gmail.com

Шергин Владимир Владимирович, orcid.org/0000-0001-7315-1625, докт. экон. наук,
Ивановский государственный химико-технологический университет
Шереметевский пр., 7, г. Иваново, 153000 Россия, e-mail: shergin476ab@mail.ru

Введение

Основным математическим аппаратом для моделирования процессов движения жидкостей и газов традиционно считаются детерминированные континуальные модели в виде систем дифференциальных уравнений с частными производными. В них непрерывность времени и пространства обеспечивается использованием бесконечно малых величин. Упомянутые классические уравнения внесли неоценимый вклад в развитие теории и практики большинства технологических процессов. Однако в последние годы появились новые подходы к математическому моделированию физико-химических явлений и, прежде всего, использование для данных целей дискретных математических объектов. В отличие от классических уравнений здесь фигурируют физические конечно малые величины, то есть потоки и потенциалы переноса квантованы, а время и пространство – дискретны [1]. Одним из видов дискретных динамических моделей для исследования физических процессов являются клеточные автоматы [2, 3].

Система клеточных автоматов является совокупностью элементов, которые в каждый момент дискретного времени могут находиться в одном из возможных состояний. Обновление состояний элементов происходит, как правило, синхронно на каждом шаге модельного времени. Этот процесс происходит согласно локальным правилам переходов, причем новое состояние определяется предыдущими состояниями как рассматриваемого элемента, так и его ближайших соседей. Разновидностями клеточного автомата являются модели решеточных газов [4, 5].При использовании различных моделей решеточных газов часто возникают вопросы, касающиеся режимов течения, при которых использование дискретных моделей будет корректным. Другая проблема состоит в обеспечении корректного масштабного перехода от модельных дискретных параметров к общепринятым макроскопическим характеристикам.

1. HPP и LBM модели решеточных газов

Известные модели решеточных газов строятся по следующим основным принципам. Исследуемое пространство разбивается на дискретные элементы некоторой пространственной решеткой. Сплошная среда представляется заполненной гипотетическими идеальными частицами, которые могут двигаться только в направлении узлов решетки. Вид пространственной решетки, а также правила перемещения и взаимодействия частиц на ней определяются конкретной принятой моделью.

Поскольку все модели решеточных газов используют дискретное пространство, их корректность контролируется выполнением законов сохранения массы, импульса

и энергии. Кроме того, на модель накладываются требования обеспечения изотропии пространства неочевидные при его дискретности.

Существует несколько разновидностей моделей решеточных газов.

Большая группа моделей предполагает рассмотрение движения отдельных частиц, заполняющих модельное пространство. При их использовании соблюдаются следующие допущения и ограничения [6]:

1. В каждый момент дискретного модельного времени частицы могут находиться только в узлах пространственной решетки.

2. Все частицы имеют одинаковую массу, равную единице.

3. Вектор скорости каждой частицы может быть направлен только в сторону одного из соседних узлов решетки.

4. Модуль вектора скорости частицы либо равен единице, либо равен нулю (если модель предполагает наличие частиц покоя).

5. В один и тот же момент времени в каждом узле решетки не могут находиться две и более частиц с одинаковыми векторами скорости.

Одной из первых была предложена модель HPP-газа, названная так по первым буквам фамилий авторов [7].

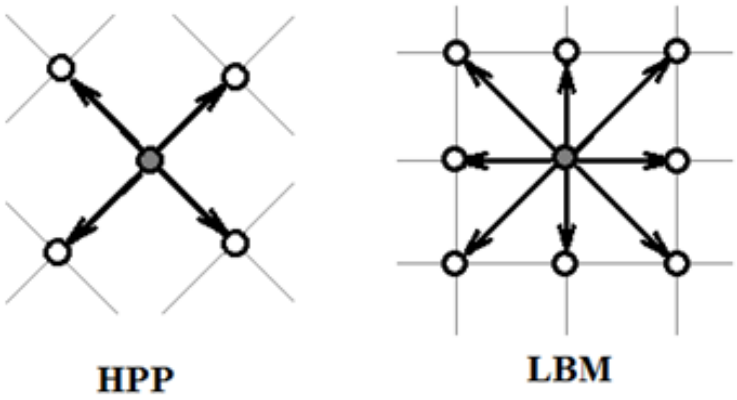


Рис. 1: Направления векторов скоростей частиц в некоторых моделях
Fig. 1: The directions of particle velocity vectors in some models

Данная модель предполагает движение виртуальных частиц между узлами ортогональной решетки. Возможные направления векторов скоростей частиц для двухмерного пространства показаны на рисунке 1 слева.

Процесс моделирования поведения среды на каждом шаге по времени разбит на два этапа: перемещение частиц в соседние узлы (этап сдвига), соударение частиц в узлах (этап столкновения). Правила поведения частиц в узлах при столкновении устанавливаются такими, чтобы несмотря на изменения векторов скоростей частиц их количество и полный импульс сохранялись. Во всех случаях процесс моделирования поведения газа представляет собой получение последовательности массивов

скоростей частиц в узлах соответствующей решетки в конкретные моменты дискретного времени [8].

Таким образом, моделируемая среда представляется эволюционирующим клеточным полем (массивом) $\Phi(t_k), k = 0, 1, ...k_{end}$, в котором каждая клетка (узел) "j" задана вектором координат r_j , а ее состояние S_j представлено булевым вектором длины b – по числу возможных направлений векторов скорости. Каждый разряд вектора определяет наличие ($s_{ji} = 1$) или отсутствие ($s_{ji} = 0$) в клетке частицы, которая движется со скоростью c_i в сторону i-того соседа:

$$\Phi = \Phi(S_j, r_j, t_k)$$

Режим работы клеточного автомата – синхронный. На этапе сдвига все частицы перемещаются в соседние узлы в направлении соответствующих векторов скоростей. На этапе взаимодействия происходит изменение состояния элементарного автомата в соответствии с принятыми в конкретной модели правилами столкновения. При этом происходит замена одного состояния клетки на другое:

$$S_j = (s_{j1}, \ldots, s_{jb}) \to S'_j(s'_{j1}, \ldots, s'_{jb}).$$

Функция переходов $\varphi(X, S)$ состоит из композиции функций сдвига φ_1 и столкновения φ_2 , то есть:

$$\varphi(X, S) = (\varphi_2 \circ \varphi_1)(X, S).$$

Функция сдвига формально может быть представлена системой подстановок, определяющих изменение компонентов вектора состояний:

$$\varphi_1 : \{s_l, r_j\}||\{\phi_l, r_j\} \to \{s'_l, r_j\}, l = 1, \ldots, b,$$

где ϕ_l - компонент вектора состояний соответствующего соседнего узла.

Функция столкновения может быть формализована подстановкой, описывающей изменение состояния клетки при взаимодействии находящихся в ней частиц:

$$\varphi_2 : \{s, r_j\} \to \{s', r_j\}.$$

Функция переходов $\varphi(X, S)$ должна обеспечивать выполнение законов сохранения. Выполнение закона сохранения массы обеспечивается равенством числа единичных компонентов векторов состояний S и S':

$$\sum_{i=1}^{b} s_i = \sum_{i=1}^{b} s'_i.$$

Для выполнения закона сохранения импульса необходимо, чтобы суммы векторов скоростей частиц в состояниях S и S' были одинаковы:

$$\sum_{i=1}^{b} c_i = \sum_{i=1}^{b} c'_i$$

Можно привести некоторые результаты моделирования течения сплошных сред с применением HPP-модели. В частности, на рисунке 2 представлены векторы скоростей потока сплошной среды в отдельных узлах решетки в установившемся режиме.

Использовалась ортогональная решетка размером 800 на 200 узлов. На левой границе рассматриваемой области располагались узлы-источники, генерирующие появление частиц, вектор скорости которых направлен внутрь области. На правой границе были расположены узлы-стоки, которые поглощали попадающие в них частицы. Плотность среды (а в конечном итоге и давление) моделировалась величиной исходной относительной концентрации виртуальных частиц, которая для приведенного случая была равна 0,1. Для устранения статистического шума проводилось осреднение значений скорости в узлах по окрестности размером 10 узлов.

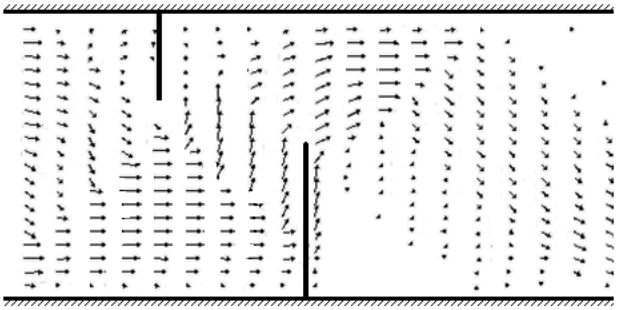


Рис. 2: Поток HPP-газа между параллельными стенками при наличии двух препятствий
Fig. 2: HPP gas flow between parallel walls, with two obstacles

Анализ полученных данных показывает, что в целом картина течения достаточно адекватна реальным представлениям. В частности, можно заметить уменьшение скорости потока у стенок, образование застойных зон и областей повышенной скорости потока при обтекании препятствий.

Продолжением и развитием дискретного подхода к моделированию течений сплошной среды стала модель, в которой используются совершенно иные представления о пространственной решетке и частицах. Модель LBM (Lattice Boltzmann Method), основана на использовании дискретного аналога уравнения Больцмана. В ней исследуется не поведение каждой частицы в отдельности, а функция распределения плотности вероятности частиц по координатам и по скоростям [9–11].

Предыдущая модель решетчатого газа, HPP, содержит много упрощений. Один из наиболее значимых недостатков состоит в том, что частицы, движущиеся по схеме столкновений, имеют большую длину свободного пробега. Данные модели хорошо подходят для моделирования газа. Но даже когда длина свободного пробега сводится к одной клетке, это не всегда дает возможность успешно моделировать жидкость [12]. В LBM понятие конкретной частицы заменили плотностью распределения частиц, а столкновения описываются оператором столкновений.

Модель LBM для 2D пространства предусматривает девять направлений скорости, которые показаны на рисунке 1 справа (модуль вектора скорости c_0 равен нулю – частицы неподвижны). Поскольку рассматривается не непосредственное движение частиц, а их распределение, то симметрия дискретного пространства достигается путем введения определенных коэффициентов в плотность распределения.

Так же, как и в модели HPP, здесь используются узлы-стенки и узлы-источники (стоки) частиц.

Итерацию клеточного автомата, функционирующего по методу LBM, можно записать следующим образом:

$$f_i(\mathbf{r} + \mathbf{c}_i, t + 1) - f_i(\mathbf{r}, t) = \Omega_i(f), \tag{1}$$

где f_i – плотность распределения частиц в направлении вектора $\mathbf{c}_i$,$\mathbf{r}$ – координата ячейки, t – время, Ω_I – оператор столкновений.

Выражение (1) является дискретным аналогом уравнения Больцмана, что было доказано [12].

Оператор столкновений вычисляется следующим образом:

$$\Omega_i(f) = -\frac{1}{\tau}\left[f_i(\mathbf{r}, t) - f_i^{eq}(\mathbf{r}, t)\right],$$

где τ – время релаксации; f_i^{eq} – функция равновесного распределения, зависящая от локальной скорости и плотности среды.

Время релаксации связано с вязкостью жидкости ν следующей формулой [13]:

$$\nu = \frac{\tau - 0,5}{3}.$$

Равновесная функция f_i^{eq} вычисляется следующим образом:

$$f_i^{eq} = w_i \rho \left[1 + 3\mathbf{c}_i\mathbf{u} + \frac{9}{2}(\mathbf{c}_i\mathbf{u})^2 - \frac{3}{2}\mathbf{u}^2\right],$$

где w_i – специальные множители; ρ и u – локальная плотность и скорость потока для ячейки соответственно.

Специальные множители выбираются таким образом, чтобы получаемые моменты импульса соответствовали моментам импульса по распределению Максвелла-Больцмана [10, 14].

Локальная плотность потока для ячейки вычисляется как сумма значений плотности распределения по всем скоростным каналам:

$$\rho = \sum f_i(\mathbf{r}, t).$$

Вектор локальной скорости для ячейки потока:

$$\mathbf{u} = \frac{1}{\rho}\sum \mathbf{c}_i f_i(\mathbf{r}, t).$$

Результат моделирования потока в объекте с неподвижными препятствиями, полученный с использованием метода LBM, показан на рисунке 3. Здесь была принята решетка размером 1000×500. Поскольку данный метод использует плотность распределения, то величина скорости на фрагменте слева закодирована оттенками черного цвета (увеличение скорости – увеличение густоты цвета). Фрагмент справа представляет распределение скоростей в сечении А-А. Поток движется слева направо.

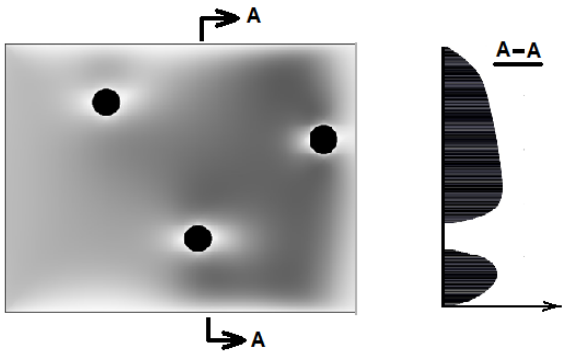


Рис. 3: Модель потока сплошной среды, полученная методом LBM
Fig. 3: Continuous flow model obtained by the LBM method

Данная картина потока весьма реалистична. Можно увидеть торможение среды у неподвижных поверхностей, наличие застойных зон. В целом, полученные результаты позволяют рекомендовать использование дискретных моделей для исследования течений жидкости или газа. Однако для более корректной проверки результатов моделирования желательно указать привычные (макроскопические) параметры рассматриваемых потоков.

2. Переход от модельных параметров процесса к реальным

Все более широкое использование решеточных моделей приводит к возникновению проблемы установления полноты соответствия (адекватности) дискретных и классических континуальных описаний потоков сплошной среды. Рассмотрим сложности, которые при этом возникают. Все модели решеточных газов рассматривают процессы, протекающие с гипотетическими частицами в идеализированном мире, где время и пространство дискретны, а основные параметры могут принимать значения только из некоторого конечного разрешенного множества. В то же время традиционно течения жидкостей и газов изучались в реальных физических условиях непрерывного времени и пространства. Следовательно, необходимо найти возможность перехода от виртуальных переменных решеточного мира к действительным физическим величинам, которые имеют смысл в макроскопических объемах реальных веществ. Следует учитывать, что в решеточном дискретном пространстве рассматриваются параметры процесса, которые являются безразмерными. На самом деле, в решеточных моделях длина измеряется количеством шагов решетки, а величина шага единична и не имеет размерности. Модельное время тоже изменяется фиксированными безразмерными единичными шагами. Отсюда следует, что скорость частиц, модельная кинематическая вязкость и ряд других параметров также не имеют размерности. В то же время в реальном пространстве рассматриваемые физические величины имеют размерность. Такая ситуация усложняет решение вопроса установления соответствия модельных и реальных процессов.

Для перехода от безразмерных модельных значений параметров к размерным

физическим значениям можно использовать следующие зависимости, в которых индекс “mod” указывает на принадлежность параметра к модельному решеточному пространству:

$$L = \frac{1}{m} \cdot L_{mod}, \tag{2}$$

$$t = \frac{1}{s} \cdot t_{mod}, \quad u = \frac{s}{m} \cdot u_{mod}, \quad \nu = \frac{s}{m^2} \cdot \nu_{mod}, \tag{3}$$

где m - число ячеек, содержащееся в одном метре, s - число модельных шагов по времени в секунду.

При использовании приведенных выше зависимостей наибольшую сложность вызывает определение модельного значения кинематической вязкости. Существует несколько подходов определения данного параметра. В литературе можно встретить выражения для расчета модельной вязкости, полученные с использованием преобразований Галилея при переходе от модельного пространства к реальному [15]. В целом, можно привести следующее уравнение, выведенное в предположении об изотропии пространства и учитывающее его дискретность. Для двухмерного пространства оно имеет вид:

$$\nu_{mod} = \frac{1}{2bd(1-2d)}, \tag{4}$$

где b – количество направлений векторов скорости частиц; d – средняя плотность частиц в расчете на одно направление скорости.

Для установления аналогий между реальными и модельными процессами течения сплошных сред можно использовать элементы теории подобия, в частности, применяя критерий гидродинамического подобия Рейнольдса (Re). Удобство его использования обусловлено тем, что поведение жидкостей и газов зависит не столько от их реальных свойств (плотности, вязкости, скорости и пр.), сколько от соотношения указанных величин. Справедливо предположить, что модельные безразмерные потоки на решетке будут подобны реальным потокам в случае их геометрического подобия и равенства критериев Рейнольдса. Если записать критерий Рейнольдса для реальной и модельной сред, получим:

$$Re = \frac{uL}{\nu}, \quad Re_{mod} = \frac{u_{mod}L_{mod}}{\nu_{mod}}. \tag{5}$$

Поскольку в гидравлически подобных системах $Re = Re_{mod}$, то можно записать следующее:

$$\frac{u}{u_{mod}} = \frac{\nu}{\nu_{mod}} \cdot \frac{L_{mod}}{L}. \tag{6}$$

Выражения (2)-(6) позволяют осуществлять переход от модельных параметров процесса к реальным (физическим) и обратно. Рассмотрим реальные условия, соответствующие потоку HPP-газа, изображенному выше на рисунке 2. Вычислительные эксперименты позволили установить, что средняя скорость среды в ядре потока была равна 0,3 м/с. Примем, что физической средой является воздух при нормальных условиях, который движется между параллельными стенками, расположенными на расстоянии 10 мм. Расчеты показывают, что для указанного случая реальная средняя скорость среды равна 0,013 м/с. Число Рейнольдса в данных условиях – 9,4.

Рассчитанные физические условия для потока, показанного на рисунке 3, следующие. Моделировалось течение воды при расстоянии между стенками, равном 50 мм. Средняя скорость потока составила 0,15 м/с, число Рейнольдса – 250.

Нами был также рассмотрен другой путь перехода от модельных параметров к реальным. Он состоит в следующем. Поскольку модель HPP рассматривает движение частиц между узлами решетки и их столкновения, можно при имитации процесса подсчитать модельную длину свободного пробега частиц и число их столкновений в течение шага дискретного времени. Далее по заданному характерному линейному размеру с помощью уравнения (6) нетрудно определить реальную (физическую) длину свободного пробега. Найденная таким образом физическая величина позволяет по справочным данным найти реальное давление конкретного газа и его кинематическую вязкость.

Обобщая результаты экспериментов с моделью HPP, проведенных при изменении плотности в диапазоне $0{,}02 \div 0{,}16$, можно указать, что реальное давление газа в опытах составляло $8 \div 80$ Па , что соответствует области среднего вакуума. При этом во всех сериях опытов значение числа Рейнольдса лежало в диапазоне от 4 до 45, что для сплошной среды должно указывать на ламинарный режим течения. Пониженное давление газа требует привлечения для анализа его течения не только числа Рейнольдса, но и числа Кнудсена (Kn), которое составляло от 0,14 до 0,01. В то же время известно, что при $10^{-3} < \mathrm{Kn} < 1$, реализуются режимы течения промежуточные между свободномолекулярным и континуальным поведением газа [16]. Рассматривая именно промежуточный режим течения, можно утверждать, что использованная в вычислительных экспериментах HPP модель решеточного газа может давать достаточно адекватные результаты, по крайней мере, для газов при пониженном давлении.

Заключение

Проведенный анализ поведения моделей решеточных газов показал следующее. Модели, рассматривающие столкновения частиц (HPP) применимы для описания течений газов при относительно невысоких значениях числа Рейнольдса, особенно при пониженном давлении. Для исследования течений реальных жидкостей более приемлемой может быть модель, использующая метод LBM.

Список литературы

[1] Wolfram S., *A new kind of science*, Wolfram media Champaign, IL, 2002.

[2] Toffoli T., Margolus N., *Cellular Automata Machines:a new environment for modeling*, Cambridge, Massachusetts: The MIT Press, 1987.

[3] Бандман О. Л., “Клеточно-автоматные модели пространственной динамики”, *Системная информатика*, **10** (2006), 59–113; [Bandman O. L., “Kletochno-avtomatnye modeli prostranstvennoj dinamiki”, *Sistemnaya informatika*, **10** (2006), 59–113, (in Russian).]

[4] Wolfram S., “Cellular automaton fluids 1: Basic theory”, *Stat. Phys.*, **45**:3-4 (1986), 471–526.

[5] Clavin P. et al, “Simulation of free boundaries in flow system by lattice-gas models”, *Journal of Fluid Mechanics*, **188** (1988), 437–464.

[6] Frish U., Hasslacher B., Pomeau Y., "Lattice-gas automata for the Navier-Stokes equation", *Physical Review Letters*, **56**:14 (1986), 1505–1508.

[7] Hardy J., de Pazzis O., Pomeau Y., "Molecular dynamics of a classical lattice gas: transport properties and time correlation functions", *Physical Review*, **13**:5 (1976), 1949–1961.

[8] Бандман О. Л., "Дискретное моделирование физико-химических процессов", *Прикладная дискретная математика*, **3** (2009), 33–49; [Bandman O. L., "Diskretnoe modelirovanie fiziko-himicheskih processov", *Prikladnaya diskretnaya matematika*, **3** (2009), 33–49, (in Russian).]

[9] Wolfram S., "Statistical mechanics of cellular automata", *Reviews of Modern Physics*, **5**:3 (1983), 601–644.

[10] Wolf-Gladrow D., *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*, Springer, 2005.

[11] Chopard B. et al., "Cellular automata and lattice Boltzmann techniques: an approach to model and simulate complex systems", *Advances in Complex Systems*, **5**:2 (2002), 103–246.

[12] Shan X., Chen H., "Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation", *Phys. Rev. E*, **49**:4 (1994), 2941–2948.

[13] He X., Luo L., "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation", *Phys. Rev. E*, **56**:6 (1997), 6811–6817.

[14] Nourgaliev R. R et al., "The lattice Boltzmann equation method: Theoretical interpretation, numerics and implications", *Int. J. Multiphase Flow*, **29**:1 (2003), 117–169.

[15] Frish U., Crutchfild J. P., Hasslacher B., "Lattice Gas hydrodynamics in two and three dimensions", *Complex Systems*, **1** (1987), 649–707.

[16] Абрамович Г. Н., *Прикладная газовая динамика. Часть 2*, М.:Наука, 1991; [Abramovich G. N., *Prikladnaya gasovaya dinamika. Chastj 2*, M.: Nauka, 1991, (in Russian).]

Bobkov S. P., Chernyavskaya A. S, Shergin V. V., "Analysis of Practical Applications of Lattice Gas Models", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 256–266.

DOI: 10.18255/1818-1015-2019-2-256-266

Abstract. In recent years, discrete approaches have been widely used in mathematical modeling of physicochemical processes. Cellular automata-based methods greatly simplify modeling procedures in many cases. In particular, this is important when using models in the form of partial differential equations systems to analyze the transfer of a substance in inhomogeneous media. In some cases, it is quite difficult to set the boundary conditions correctly if the object of study has boundaries of complex shape. It is also difficult to use mathematical physics classical equations if one cannot neglect the influence of stochastic effects on the process flow. The lattice gas models considered in the article are one of the types of cellular automata. Until now they have not been widely adopted, despite the fact that the first works on their use appeared about forty years ago. It is known, however, that lattice gases successfully describe a number of hydrodynamic phenomena, and the results obtained do not contradict the generally accepted views on the physical nature of continuous media motion processes. When using models of lattice gases, there are often questions about the correctness of the use of discrete models in various flow regimes. The second problem is a large-scale transition from model discrete parameters to generally accepted macroscopic characteristics of flows, such as flow velocity, viscosity and density of the medium, etc. It is also necessary to take into account that the indicated parameters in the lattice model are dimensionless, and the corresponding real macroscopic parameters have dimension. In this paper, an attempt is made to propose a method of large-scale transition, as well as to indicate the areas of practical use of some models of lattice gases.

Keywords: discrete approach, lattice gas, HPP, FHP and LBM models

On the authors:
Sergey P. Bobkov, orcid.org/0000-0001-7315-1625, PhD,
P.G. Ivanovo State University of Chemistry and Technology,
7 Sheremetevsky str., Ivanovo 153000, Russia, e-mail: bsp@isuct.ru

Anastasiya S. Chernyavskaya, orcid.org/0000-0001-9717-9665, assistant,
Ivanovo State University of Chemistry and Technology,
7 Sheremetevsky str., Ivanovo 153000, Russia, e-mail: olga@mail.ru

Vladimir V. Shergin, orcid.org/0000-0001-9518-8058, PhD,
Ivanovo State University of Chemistry and Technology,
7 Sheremetevsky str., Ivanovo 153000, Russia, e-mail: shergin476ab@mail.ru

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 267–278
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 267–278

DOI: 10.18255/1818-1015-2019-2-267-278

УДК 519.2

Существование несмещенной состоятельной оценки энтропии для специальной меры Бернулли

Тимофеев Е. А.

Поступила в редакцию 6 мая 2019
После доработки 22 мая 2019
Принята к публикации 24 мая 2019

Аннотация.

Пусть $\Omega = \mathcal{A}^{\mathbb{N}}$ – пространство правосторонних бесконечных последовательностей символов из алфавита $\mathcal{A} = \{0, 1\}$, $\mathbb{N} = \{1, 2, \dots\}$,

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{\infty} |x_k - y_k| 2^{-k}$$

– метрика на Ω, и μ – вероятностная мера на Ω. Пусть $\boldsymbol{\xi_0}, \boldsymbol{\xi_1}, \dots, \boldsymbol{\xi_n}$ – независимые случайные точки на Ω, распределенные по мере μ. Будем изучать оценку $\eta_n^{(k)}(\gamma)$ – величину обратной к энтропии $1/h$, которая определяется следующим образом.

$$\eta_n^{(k)}(\gamma) = k \left(r_n^{(k)}(\gamma) - r_n^{(k+1)}(\gamma) \right),$$

где

$$r_n^{(k)}(\gamma) = \frac{1}{n+1} \sum_{j=0}^{n} \gamma \left(\min_{i:i \neq j}{}^{(k)} \rho(\boldsymbol{\xi_i}, \boldsymbol{\xi_j}) \right),$$

$\min^{(k)}\{X_1, \dots, X_N\} = X_k$, если $X_1 \leq X_2 \leq \dots \leq X_N$. Число k и функция $\gamma(t)$ – вспомогательные параметры. Основной результат работы

Теорема. Пусть μ – мера Бернулли с вероятностями $p_0, p_1 > 0$, $p_0 + p_1 = 1$, $p_0 = p_1^2$, тогда $\forall \varepsilon > 0$ существует непрерывная функция $\gamma(t)$ такая, что

$$\left| \mathsf{E}\eta_n^{(k)}(\gamma) - \frac{1}{h} \right| < \varepsilon, \quad \mathsf{Var}\ \eta_n^{(k)}(\gamma) \to 0,\ n \to \infty.$$

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

Для цитирования: Тимофеев Е. А., "Существование несмещенной состоятельной оценки энтропии для специальной меры Бернулли", *Моделирование и анализ информационных систем*, **26**:2 (2019), 267–278.

Об авторах:
Тимофеев Евгений Александрович, orcid.org/0000-0002-3094-4390, доктор. физ.-мат. наук., профессор кафедры теоретической информатики,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: timofeevEA@gmail.com

В работе [4] предложена непараметрическая оценка энтропии, которая зависит от семейства метрик. В [7] эта оценка модифицирована так, что она зависит от конкретной метрики и некоторой функции $\gamma(t)$. Эта модификация более удобна для анализа, поэтому именно она и будет рассматриваться в настоящей работе.

Основным показателем качества оценки является ее точность, которая складывается из несмещенности и дисперсии оценки. Поиск порядка убывания дисперсии оценки является более простой задачей. Почти оптимальный порядок убывания дисперсии доказан в [5]. Нахождение математического ожидания оценки в общем случае является очень трудной задачей, найти решение которой, по-видимому, невозможно. Поэтому находить ее будем для мер Бернулли. Для мер Бернулли асимптотическая несмещенность доказана для почти всех значений параметров и конкретной функции $\gamma(t)$ [8]. Для бинарных мер Бернулли с параметрами p_0, p_1, $(p_0 + p_1 = 1)$ исключительные случаи, в которых асимптотическая сходимость не доказана, описываются как

$$\frac{\log p_0}{\log p_1} - \text{рационально.}$$

Отсутствие сходимости в исключительных случаях является принципиальной трудностью, поскольку при простейшем выборе метрики и функции $\gamma(t)$ в случае $p_0 = p_1 = 1/2$ показано [6], что оценка является смещенной, хотя смещение очень мало (порядка 10^{-6}).

В [4] показано, что при метрике, заданной в (1), и функции $\gamma(t) = -\log_2 t$ в случае $p_0 = p_1 = 1/2$, оценка будет несмещенной.

В [10] показано, что и для следующего исключительного случая

$$\frac{\log p_0}{\log p_1} = 2,$$

для метрики, заданной в (1), можно найти такую функцию $\gamma(t)$, для которой оценка энтропии будет несмещенной. Однако найденная функция $\gamma(t)$ является не ограниченной.

В настоящей работе будет показано, что для этого исключительного случая и для метрики, заданной в (1), можно найти такую непрерывную монотонную функцию $\gamma(t)$, для которой оценка энтропии будет несмещенной, а из непрерывности и монотонности этой функции следует и состоятельность оценки [4].

Таким образом, предложенный в [4, 7] метод адаптивного оценивания энтропии (предварительное нахождение вспомогательной функции) работает и в исключительных случаях.

1. Непараметрическая оценка энтропии

Обозначим через $\Omega = \mathcal{A}^{\mathbb{N}}$ пространство правосторонних бесконечных последовательностей символов из конечного алфавита $\mathcal{A} = \{0, 1\}$.

Пусть даны $\boldsymbol{\xi_0}, \boldsymbol{\xi_1}, \ldots, \boldsymbol{\xi_n}$ – независимые случайные точки в Ω, одинаково распределенные по вероятностной мере μ.

Зададим метрику ρ на Ω, положив

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{\infty} |x_i - y_i| 2^{-i}. \tag{1}$$

Пусть заданы:

1. k – вспомогательный параметр, который будем считать небольшим и фиксированным. Этот параметр служит для контроля применимости (оценки, полученные для различных значений k, являются оценками одной и той же величины);

2. $\gamma(t)$ – функция, определенная на полуинтервале $(0, 1]$.

Оценка $\eta_n^{(k)}(\gamma)$ величины обратной к энтропии $1/h$ определяется следующим образом:

$$\eta_n^{(k)}(\gamma) = k \left(r_n^{(k)}(\gamma) - r_n^{(k+1)}(\gamma) \right), \tag{2}$$

где

$$r_n^{(k)}(\gamma) = \frac{1}{n+1} \sum_{j=0}^{n} \gamma \left(\min_{i:i\neq j}{}^{(k)} \rho(\boldsymbol{\xi_i}, \boldsymbol{\xi_j}) \right), \tag{3}$$

и $\min^{(k)}\{X_1, \ldots, X_N\} = X_k$, если $X_1 \leq X_2 \leq \cdots \leq X_N$.

Подчеркнем, что логарифм в определении энтропии выбирается натуральным (в этом случае в формуле (3) нет дополнительных множителей).

Для дисперсии статистики $r_n^{(k)}(\gamma_0)$ в [5] показано, что $\mathsf{Var}\ r_n^{(k)}(\gamma_0) = \mathcal{O}(n^{-1} \ln n)$.

В работе [4] аналогичные оценки доказаны для произвольной монотонной функции $\gamma(t)$, имеющей порядок $\log_2 t$.

Смещение оценки $\eta_n^{(k)}(\gamma_0)$ изучено в [8], где доказана асимптотическая несмещенность, если логарифмы некоторых вероятностей рационально не соизмеримы. Для меры Бернулли с вероятностями $p_0, p_1, \ldots, p_m$ асимптотическая несмещенность выполняется, если $\log p_0, \log p_1, \ldots, \log p_m$ рационально не соизмеримы.

2. Построение функции $\gamma(t)$ при $p_0 = p_1^2$

Теорема 1. *Пусть μ – мера Бернулли с параметрами $p_0 = q^2$, $p_1 = q$, тогда $\forall \varepsilon > 0$ существует непрерывная функция $\gamma(t)$ такая, что*

$$\left| \mathsf{E}\eta_n^{(k)}(\gamma) - \frac{1}{h} \right| < \varepsilon, \quad \mathsf{Var}\ \eta_n^{(k)}(\gamma) \to 0,\ n \to \infty.$$

Доказательство. Для доказательства теоремы достаточно показать [4, 9], что $\forall \varepsilon > 0$ существует непрерывная монотонная функция $\gamma(t)$ такая, что в точках некоторого ε-разбиения отрезка $[0, 1]$

$$\chi(t) = -\frac{\ln t}{h}, \tag{4}$$

где

$$\chi(t) = \int_{\Omega} \gamma(\nu(t, \omega))\, d\mu(\omega), \tag{5}$$

а через $r = \nu(t, \omega)$ обозначается обратная функция к мере шара $B(r, \omega)$ радиуса r с центром в точке ω.

Для нахождения решения уравнения (4) введем вспомогательные функции

$$\chi_{n,j}(t) = \int_\Omega \gamma(j2^{-n} + 2^{-n}\nu(t,\omega))\, d\mu(\omega), \quad 0 \le j < 2^n, \quad n = 0, 1, 2, \ldots \tag{6}$$

и будем считать, что $\chi_{0,0}(t) = \chi(t)$.

Для метрики (1) и меры Бернулли с параметрами $p_0 = q^2$, $p_1 = q$ функция $r = \nu(t, \omega)$ удовлетворяет рекуррентному уравнению:

$$\begin{gathered}\nu(t, 0\omega) = \begin{cases} \frac{1}{2}\nu\left(\frac{t}{q^2}, \omega\right), & 0 \le t \le q^2; \\ \frac{1}{2} + \frac{1}{2}\nu\left(\frac{t-q^2}{q}, \omega\right), & q^2 \le t \le 1; \end{cases} \\ \nu(t, 1\omega) = \begin{cases} \frac{1}{2}\nu\left(\frac{t}{q}, \omega\right), & 0 \le t \le q; \\ \frac{1}{2} + \frac{1}{2}\nu\left(\frac{t-q}{q^2}, \omega\right), & q \le t \le 1. \end{cases}\end{gathered} \tag{7}$$

Подставляя (7) в (6), получим

$$\chi_{n,j}(t) = \begin{cases} q^2\chi_{n+1,2j}\left(\dfrac{t}{q^2}\right) + q\chi_{n+1,2j}\left(\dfrac{t}{q}\right), & 0 \le t \le q^2; \\ q^2\chi_{n+1,2j+1}\left(\dfrac{t-q^2}{q}\right) + q\chi_{n+1,2j}\left(\dfrac{t}{q}\right), & q^2 \le t \le q; \\ q^2\chi_{n+1,2j+1}\left(\dfrac{t-q^2}{q}\right) + q\chi_{n+1,2j+1}\left(\dfrac{t-q}{q^2}\right), & q \le t \le 1. \end{cases} \tag{8}$$

В [10] показано, что для нахождения решения уравнения (4) достаточно найти решение этого уравнения с непрерывной монотонной функцией $\chi(t)$ с условием $\chi(0) = 0$, $\chi(1) = 1$. В этом случае функция $\chi(t)$ является функцией распределения некоторой случайной величины.

Лемма 1. *Пусть для некоторых величин $A_{n,0} \ge 0$, $\sum_{n=0}^{\infty} A_{n,0} = 1$ функция $\gamma(t)$ удовлетворяет равенствам:*

$$\gamma(2^{-n-1}t + 2^{-n-1}) = A_{n,0}\gamma(t) + 1 - \sum_{k=0}^{n} A_{k,0}, \quad n = 0, 1, \ldots, \infty, \tag{9}$$

тогда

$$\begin{array}{ll} \chi(q + q^2 t) & = A_{0,0}\left[q\chi(t) + q^2\chi(q^2 + qt)\right] + 1 - A_{0,0}; \\ \chi(q^{m+1} + q^{m+2}t) & = q^2 A_{0,m-1}\chi(q^2 t) + A_{0,m}\left[q\chi(t) + q^2\chi(q^2 + qt)\right] + \\ & +1 - \sum_{k=0}^{m} A_{0,k}, \quad m = 1, 2, \ldots, \infty; \end{array} \tag{10}$$

где

$$A(qy + q^2y^2, 0) = A(0, y), \tag{11}$$

а через $A(x, y)$ обозначается производящая функция:

$$A(x, y) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} A_{n,m}x^n y^m. \tag{12}$$

Доказательство. Подчеркнем, что функции $\gamma(t)$ и $\chi(t)$ задают инвариантные меры для итерационной системы подобий с вероятностями. Существование таких мер следует из известной теоремы Хатчинсона [2], [1] и ее обобщения для бесконечной системы подобий [3]. Отметим, что для функции $\chi(t)$ итерационная система подобий имеет перекрытия.

Покажем, что функции $\chi_{n,0}(t)$ удовлетворяют системе уравнений:

$$\begin{aligned} \chi_{n,0}(q+q^2t) &= A_{n,0}\left[q\chi(t)+q^2\chi(q^2+qt)\right]+B_{n,0};\\ \chi_{n,0}(q^{m+1}+q^{m+2}t) &= q^2A_{n,m-1}\chi(q^2t)+A_{n,m}\left[q\chi(t)+q^2\chi(q^2+qt)\right]+B_{n,m},\\ &\quad m=1,2,\ldots,\infty; \end{aligned} \tag{13}$$

где

$$\begin{aligned} &A_{n,1}=qA_{n+1,0},\\ &A_{n,m}=q^2A_{n+1,m-2}+qA_{n+1,m-1},\\ &B_{n,0}=1-\textstyle\sum_{k=0}^{n}A_{k,0},\\ &B_{n,1}=qB_{n+1,0}+q^2B_{n,0},\\ &B_{n,m}=q^2B_{n+1,m-2}+qB_{n+1,m-1},\quad m=2,3,\ldots,\infty. \end{aligned} \tag{14}$$

Подставляя (9) в (6), получим

$$\chi_{n+1,1}(t)=A_{n,0}\chi(t)+B_{n,0}, \tag{15}$$

где $B_{n,0}$ задано в (14).

Подставляя (15) в третье уравнение (8), получим первое уравнение в (13).

Из второго уравнения (8) имеем

$$\begin{aligned} \chi_{n,0}(q^2+q^3t) &= q^2\chi_{n+1,1}(q^2t)+q\chi_{n+1,0}(q+q^2t)=\\ &= q^2A_{n,0}\chi(q^2t)+q^2B_{n,0}+qA_{n+1,0}\left[q\chi(t)+q^2\chi(q^2+qt)\right]+qB_{n+1,0}. \end{aligned}$$

Используя первое и четвертое равенство в (14), получим второе уравнение в (13) при $m=1$.

Из первого уравнения (8) имеем

$$\begin{aligned} \chi_{n,0}(q^{m+1}+q^{m+2}t) &= q^2\chi_{n+1,1}(q^{m-1}+q^mt)+q\chi_{n+1,0}(q^m+q^{m+1}t)=\\ &= q^4A_{n+1,m-3}\chi(q^2t)+q^2A_{n+1,m-2}\left[q\chi(t)+q^2\chi(q^2+qt)\right]+q^2B_{n+1,m-2}+\\ &\quad +q^3A_{n+1,m-2}\chi(q^2t)+qA_{n+1,m-1}\left[q\chi(t)+q^2\chi(q^2+qt)\right]+qB_{n+1,m-1}. \end{aligned}$$

Используя второе и пятое равенство в (14), получим второе уравнение в (13) при $m>1$.

Из уравнений (14) найдем уравнение для производящей функции (12).

$$A(x,y)-A(x,0)=qx^{-1}y\left(A(x,y)-A(0,y)\right)+q^2x^{-1}y^2\left(A(x,y)-A(0,y)\right).$$

Следовательно,

$$A(x,y)\left(x-qy-q^2y^2\right)=xA(x,0)-(qy+q^2y^2)A(0,y).$$

При $x=qy+q^2y^2$ получаем (11).

□

Лемма 2. *Пусть для некоторых величин* $A_{0,m} \geq 0$, $\sum_{m=0}^{\infty} A_{0,m} = 1$ *функция* $\chi(t)$ *удовлетворяет равенствам* (10), *пусть производящая функция* $A(x, y)$ *удовлетворяет уравнению* (11) *и коэффициенты* $A_{n,0} \geq 0$, *тогда функция* $\gamma(t)$ *удовлетворяет уравнению* (9).

Доказательство. Индукцией по n покажем, что функции $\chi_{n,0}(t)$ удовлетворяют системе уравнений (13), функции $\chi_{n,1}(t)$ – уравнению (15), а коэффициенты $A_{n,m}$, $B_{n,m}$ – уравнениям (14).

При $n = 0$ функции $\chi_{n,0}(t)$ удовлетворяют системе уравнений (13) по условию леммы. Предположим, что условия выполнены для некоторого n и покажем, что они выполнены для $n + 1$.

Применим Лемму 1 из работы [10], в которой утверждается, что для ограниченной функции $\chi_{n+1,1}(t)$

$$\chi_{n+1,1}(t) = \sum_{k=0}^{\infty} (-1)^k q^{k-1} \chi_{n,0}\left(1 - q^{k+2}(1-t)\right), \quad 0 \leq t \leq 1. \tag{16}$$

Поскольку $1 - q^{k+2}(1-t) \geq q$, то подставляя первое равенство из (13), получим

$$\begin{aligned}
\chi_{n+1,1}(t) &= B_{n,0} + \\
&+ A_{n,0} \sum_{k=0}^{\infty} (-1)^k q^{k-1} \left[q\chi_{n,0}\left(1 - q^k(1-t)\right) + q^2 \chi_{n,0}\left(1 - q^{k+1}(1-t)\right)\right] = \\
&= B_{n,0} + A_{n,0} \left[\sum_{k=0}^{\infty} (-1)^k q^k \chi_{n,0}\left(1 - q^k(1-t)\right) + \sum_{k=1}^{\infty} (-1)^k q^k \chi_{n,0}\left(1 - q^k(1-t)\right)\right] = \\
&= A_{n,0}\chi(t) + B_{n,0}.
\end{aligned}$$

Следовательно, получаем, что функция $\gamma(t)$ удовлетворяет уравнению (9).

Покажем, что функции $\chi_{n+1,0}(t)$ удовлетворяют системе уравнений (13).

Из второго уравнения (8) при $j = 0$ имеем

$$\begin{aligned}
\chi_{n+1,0}(q + q^2 t) &= q^{-1}\chi_{n,0}(q^2 + q^3 t) - q\chi_{n+1,1}(q^2 t) = \\
&= q^{-1} A_{n,1} \left[q\chi(t) + q^2 \chi(q^2 + qt)\right] + q^{-1} B_{n,1} - qB_{n,0}.
\end{aligned}$$

Используя первое и четвертое равенство в (14), получим первое уравнение в (13) при $n + 1$.

Из первого уравнения (8) при $j = 0$ имеем

$$\begin{aligned}
\chi_{n+1,0}(q^{m+1} + q^{m+2} t) &= q^{-1}\chi_{n,0}(q^{m+2} + q^{m+3} t) - q\chi_{n+1,0}(q^m + q^{m+1} t) = \\
&= qA_{n,m}\chi(q^2 t) + q^{-1} A_{n,m+1} \left[q\chi(t) + q^2 \chi(q^2 + qt)\right] + q^{-1} B_{n,m+1} - \\
&\quad - q^3 A_{n+1,m-2}\chi(q^2 t) - qA_{n+1,m-1} \left[q\chi(t) + q^2 \chi(q^2 + qt)\right] - qB_{n+1,m-1}.
\end{aligned}$$

Используя второе и пятое равенство в (14), получим второе уравнение в (13) при $n + 1$.

□

Подчеркнем, что в отличие от предыдущей леммы, условие $A_{n,0} \geq 0$ здесь существенно и трудно проверяемо. Действительно, уравнение (11) можно переписать как

$$A(x,0) = A(0, q^{-1}xC(-x)), \tag{17}$$

где

$$C(x) = \frac{1-\sqrt{1-4x}}{2x} = \sum_{k=0}^{\infty} \frac{1}{k}\binom{2k}{k} x^k.$$

– функция Каталана.

Приведем еще один класс непрерывных функций, которые реализуются некоторой функцией $\gamma(t)$.

Лемма 3. *Пусть для некоторых величин $A_{n,0} \geq 0$, $\sum_{n=0}^{\infty} A_{n,0} = 1$ функция $\gamma(t)$ удовлетворяет равенствам*

$$\gamma(2^{-n-1}t + 1 - 2^{-n}) = A_{n,0}\gamma(t) + \sum_{k=0}^{n-1} A_{k,0}, \quad n = 0, 1, \ldots, \infty, \tag{18}$$

тогда

$$\begin{array}{ll} \chi(q^2t) & = A_{0,0}\left[q^2\chi(t) + q\chi(qt)\right]; \\ \chi(1-q^m+q^{m+2}t) & = qA_{0,m-1}\chi(q+q^2t) + A_{0,m}\left[q^2\chi(t) + q\chi(qt)\right] + \\ & +q^2A_{0,m-1} + \sum_{k=0}^{m-2} A_{0,k}, \quad m = 1, 2, \ldots, \infty; \end{array} \tag{19}$$

где

$$A(q^2y + qy^2, 0) = A(0, y), \tag{20}$$

где через $A(x,y)$ обозначается производящая функция.

Доказательство. Покажем, что функции $\chi_{n,2^n-1}(t)$ удовлетворяют системе уравнений

$$\begin{array}{ll} \chi_{n,2^n-1}(q^2t) & = A_{n,0}\left[q^2\chi(t) + q\chi(qt)\right] + B_{n,0}; \\ \chi_{n,2^n-1}(1-q^m+q^{m+2}t) & = qA_{n,m-1}\chi(q+q^2t) + A_{n,m}\left[q^2\chi(t) + q\chi(qt)\right] + B_{n,m}, \\ & m = 1, 2, \ldots, \infty; \end{array} \tag{21}$$

где

$$\begin{array}{l} A_{n,1} = q^2A_{n+1,0}, \\ A_{n,m} = q^2A_{n+1,m-1} + qA_{n+1,m-2}, \\ B_{n,0} = \sum_{k=0}^{n-1} A_{k,0}, \\ B_{n,1} = q^2B_{n+1,0} + qB_{n,0}, \\ B_{n,m} = q^2B_{n+1,m-1} + qB_{n+1,m-2}, \quad m = 2, 3, \ldots, \infty. \end{array} \tag{22}$$

Подставляя (18) в (6), получим

$$\chi_{n+1,2^{n+1}-2}(t) = A_{n,0}\chi(t) + B_{n,0}, \tag{23}$$

где $B_{n,0}$ задано в (22).

Подставляя (23) в первое уравнение (8), получим первое уравнение в (21).

Из второго уравнения (8) имеем

$$\chi_{n,2^n-1}(q^2+q^3t) = q^2\chi_{n+1,2^{n+1}-1}(q^2t) + q\chi_{n+1,2^{n+1}-2}(q+q^2t) = \\ = qA_{n,0}\chi(q+q^2t) + qB_{n,0} + q^2A_{n+1,0}\left[q^2\chi(t)+q\chi(qt)\right] + q^2B_{n+1,0}.$$

Используя первое и четвертое равенство в (22), получим второе уравнение в (21) при $m=1$.

Из третьего уравнения (8) имеем

$$\chi_{n,2^n-1}(1-q^m+q^{m+2}t) = q^2\chi_{n+1,2^{n+1}-1}(1-q^{m-1}+q^{m+1}t)+q\chi_{n+1,2^{n+1}-1}(1-q^{m-2}+q^mt) = \\ = q^3A_{n+1,m-2}\chi(q+q^2t) + q^2A_{n+1,m-1}\left[q^2\chi(t)+q\chi(qt)\right] + q^2B_{n+1,m-1}+ \\ + q^2A_{n+1,m-3}\chi(q+q^2t) + qA_{n+1,m-2}\left[q^2\chi(t)+q\chi(qt)\right] + qB_{n+1,m-2}.$$

Используя второе и пятое равенство в (22), получим второе уравнение в (21) при $m>1$.

Из уравнений (22) найдем уравнение для производящей функции (12):

$$A(x,y) - A(x,0) = q^2x^{-1}y\left(A(x,y)-A(0,y)\right) + qx^{-1}y^2\left(A(x,y)-A(0,y)\right).$$

Следовательно,

$$A(x,y)\left(x-q^2y-qy^2\right) = xA(x,0) - (q^2y+qy^2)A(0,y). \tag{24}$$

При $x = q^2y+qy^2$ получаем (20).

□

Лемма 4. *Пусть для некоторых величин $A_{0,m} \geq 0$, $\sum_{m=0}^{\infty} A_{0,m} = 1$ функция $\chi(t)$ удовлетворяет равенствам (19), пусть производящая функция $A(x,y)$ удовлетворяет уравнению (24) и коэффициенты $A_{n,0} \geq 0$, тогда функция $\gamma(t)$ удовлетворяет уравнению (18).*

Доказательство. Индукцией по n покажем, что функции $\chi_{n,2^n-1}(t)$ удовлетворяют системе уравнений (21), функции $\chi_{n,2^n-2}(t)$ – уравнению (23), а коэффициенты $A_{n,m}$, $B_{n,m}$ – уравнениям (22).

При $n=0$ функции $\chi_{n,2^n-1}(t)$ удовлетворяют системе уравнений (21) по условию леммы. Предположим, что условия выполнены для некоторого n и покажем, что они выполнены для $n+1$.

Применим Лемму 1 из работы [10], в которой утверждается, что для ограниченной функции $\chi_{n+1,2^{n+1}-1}(t)$

$$\chi_{n+1,2^{n+1}-1}(t) = \sum_{k=0}^{\infty}(-1)^kq^{k-1}\chi_{n,2^n-1}\left(1-q^{k+2}(1-t)\right). \tag{25}$$

Подставляя (21), получим

$$\chi_{n+1,2^{n+1}-1}(q^2t) = \\ = \sum_{k=0}^{\infty}(-1)^k q^{k-1}\left[qA_{n,k+1}\chi(q+q^2t) + A_{n,k+2}\left(q^2\chi(t)+q\chi(qt)\right) + B_{n,k+2}\right] = \\ = \chi(q+q^2t)\sum_{k=0}^{\infty}(-1)^k q^k A_{n,k+1} + \\ + \left(q^2\chi(t)+q\chi(qt)\right)\sum_{k=0}^{\infty}(-1)^k q^{k-1}A_{n,k+2} + \\ + \sum_{k=0}^{\infty}(-1)^k q^{k-1}B_{n,k+2}.$$

Из уравнения для производящей функции (24) имеем

$$A(x,-q) = A(x,0),$$

поэтому

$$\sum_{k=0}^{\infty}(-1)^k q^k A_{n,k+1} = 0, \quad \sum_{k=0}^{\infty}(-1)^k q^{k-1}A_{n,k+2} = q^{-2}A_{n,1} = A_{n+1,0}. \tag{26}$$

Аналогично показывается, что

$$\sum_{k=0}^{\infty}(-1)^k q^{k-1}B_{n,k+2} = B_{n+1,0}.$$

Следовательно,

$$\chi_{n+1,2^{n+1}-1}(q^2t) = A_{n+1,0}\left(q^2\chi(t)+q\chi(qt)\right) + B_{n+1,0}.$$

Подставляя в (25) $t = 1-q^m+q^{m+2}t$, получим

$$\chi_{n+1,2^{n+1}-1}(1-q^m+q^{m+2}t) = \sum_{k=0}^{\infty}(-1)^k q^{k-1}\chi_{n,2^n-1}\left(1-q^{k+m+2}+q^{k+m+4}t\right).$$

Подставляя (21), получим

$$\chi_{n+1,2^{n+1}-1}(1-q^m+q^{m+2}t) = \\ = \sum_{k=0}^{\infty}(-1)^k q^{k-1}\left[qA_{n,k+m+1}\chi(q+q^2t) + A_{n,k+m+2}\left(q^2\chi(t)+q\chi(qt)\right) + B_{n,k+m+2}\right] = \\ = \chi(q+q^2t)\sum_{k=0}^{\infty}(-1)^k q^k A_{n,k+m+1} + \\ + \left(q^2\chi(t)+q\chi(qt)\right)\sum_{k=0}^{\infty}(-1)^k q^{k-1}A_{n,k+m+2} + \\ + \sum_{k=0}^{\infty}(-1)^k q^{k-1}B_{n,k+m+2}.$$

Индукцией по m покажем, что

$$\sum_{k=0}^{\infty}(-1)^k q^{k-1} A_{n,k+m+2} = A_{n+1,m}.$$

При $m = 0$ это равенство доказано в (26).

Применяя второе равенство в (22), получим

$$\sum_{k=0}^{\infty}(-1)^k q^{k-1} A_{n,k+m+2} =$$
$$= \sum_{k=0}^{\infty}(-1)^k q^{k-1} \left[q^2 A_{n+1,k+m+1} + q A_{n,k+m}\right] =$$
$$= q^2 A_{n+2,m-1} + q A_{n+2,m-2} = A_{n+1,m}.$$

Аналогично показывается, что

$$\sum_{k=0}^{\infty}(-1)^k q^{k-1} B_{n,k+m+2} = B_{n+1,m}.$$

Следовательно,

$$\chi_{n+1,2^{n+1}-1}(1 - q^m + q^{m+2}t) = qA_{n+1,m-1}\chi(q + q^2 t) + A_{n+1,m}\left(q^2\chi(t) + q\chi(qt)\right) + B_{n+1,m}.$$

Итак, функции $\chi_{n,2^n-1}(t)$ удовлетворяют системе уравнений (21).

Покажем, что, функции $\chi_{n+1,2^{n+1}-2}(t)$ удовлетворяют уравнению (23). Из второго уравнения (6), имеем

$$\chi_{n+1,2^{n+1}-2}(q + q^2 t) = q^{-1}\chi_{n,2^n-1}(q^2 + q^3 t) - q\chi_{n+1,2^{n+1}-1}(q^2 t) =$$
$$= A_{n,0}\chi(q + q^2 t) + q^{-1}A_{n,1}\left(q^2\chi(t) + q\chi(qt)\right) + q^{-1}B_{n,1} -$$
$$- qA_{n+1,0}\left(q^2\chi(t) + q\chi(qt)\right) - qB_{n+1,0} =$$
$$= A_{n,0}\chi(q + q^2 t) + \left[q^{-1}A_{n,1} - qA_{n+1,0}\right]\left(q^2\chi(t) + q\chi(qt)\right) + q^{-1}B_{n,1} - qB_{n+1,0} =$$
$$= A_{n,0}\chi(q + q^2 t) + B_{n,0}.$$

Индукцией по m покажем, что уравнение

$$\chi_{n+1,2^{n+1}-2}(t) = A_{n,0}\chi(t) + B_{n,0},$$

выполняется при $t \geq q^m$. При $m = 1$ это доказано.

Из третьего уравнения (6), имеем

$$\chi_{n+1,2^{n+1}-2}(qt) = q^{-1}\chi_{n,2^n-1}(q^2 t) - q\chi_{n+1,2^{n+1}-2}(t) =$$
$$= +q^{-1}A_{n,0}\left(q^2\chi(t) + q\chi(qt)\right) + q^{-1}B_{n,0} - qA_{n,0}\chi(t) - qB_{n,0} =$$
$$= A_{n,0}\chi(qt) + B_{n,0}.$$

Следовательно, получаем, что функция $\gamma(t)$ удовлетворяет уравнению (18).

□

Перейдем к доказательству теоремы.

Определим систему подмножеств $Q_n \subset [0,1]$ по правилу:

$$Q_1 = \{0, q^2, q, 1\};$$

$t \in Q_{n+1}$, если

$$\begin{array}{ll} \frac{t}{q^2}, \frac{t}{q} \in Q_n, & t \le q^2; \\ \frac{t-q^2}{q}, \frac{t}{q} \in Q_n, & q^2 \le t \le q; \\ \frac{t-q^2}{q}, \frac{t-q}{q^2} \in Q_n, & q \le t. \end{array}$$

Отметим, что при построении множеств можно использовать следующее свойство:

$$Q_{n+1} \subset qQ_n \cup q^2 + qQ_n.$$

Покажем, что существует функция $\chi^{(n)}(t)$ такая, что

$$\chi^{(n)}(t_k) = a_k, \quad \forall t_k \in Q_n.$$

Для построения такой функции выберем функцию $\gamma^{(n)}(t)$

$$\gamma^{(n)}(t,\omega) = \begin{cases} \alpha\gamma_0^{(n-1)}(2t), & 0 \le t \le 0.5; \\ \alpha + (1-\alpha)\gamma_1^{(n-1)}(2t-1), & 0.5 \le t \le 1; \end{cases}$$

а функции $\gamma_0^{(n-1)}(t)$, $\gamma_1^{(n-1)}(t)$ строятся по соответствующим функциям $\chi_0^{(n-1)}(t)$, $\chi_1^{(n-1)}(t)$, которые задаются на множестве Q_{n-1}.

На множестве Q_1 функции $\gamma_i^{(0)}(t)$ строятся по леммам 2, 4, причем соответствующую функцию $\chi_0^{(0)}(t)$ достаточно задать только в точке q, а функцию $\chi_1^{(0)}(t)$ – только в точке q^2.

□

Список литературы / References

[1] Falconer K. J., *Fractal geometry : Mathematical Foundation and Applications*, John Wiley & Sons, NY USA, 1990.

[2] Hutchinson J. E., “Fractals and self-similarity”, *Indiana Univ. Math. J.*, **30** (1981), 713–747.

[3] Mauldin R. D. and Williams S. C., “Random Recursive Constructions: Asymptotic Geometric and Topological Properties”, *Transactions of the American Mathematical Society*, **295**:1 (1986), 325–346.

[4] Timofeev E. A., “Selection of a Metric for the Nearest Neighbor Entropy Estimators”, *Journal of Mathematical Sciences*, **203**:6 (2014), 892–906.

[5] Kaltchenko A., Timofeeva N., “Entropy Estimators with Almost Sure Convergence and an $O(n^{-1})$ Variance”, *Advances in Mathematics of Communications*, **2**:1 (2008), 1–13.

[6] Kaltchenko A., Timofeeva N., “Rate of convergence of the nearest neighbor entropy estimator”, *AEU – International Journal of Electronics and Communications*, **64**:1 (2010), 75–79.

[7] Тимофеева Н. Е., "Построение оценки энтропии для специальной метрики и произвольной функции", *Модел. и анализ информ. систем*, **20**:6 (2013), 174-–178; [Timofeeva N. E., "Construction of Entropy Estimator with Special Metric and Arbitrary Function", *Modeling and Analysis of Information Systems*, **20**:6 (2013), 174-–178, (in Russian).]

[8] Timofeev E. A., "Bias of a nonparametric entropy estimator for Markov measures", *Journal of Mathematical Sciences*, **176**:2 (2011), 255–269.

[9] Timofeev E. A., "Statistical Estimation of measure invariants", *St.Petersburg Math. J.*, **17**:3 (2006), 527–551.

[10] Тимофеев Е. А., "Существование несмещенной оценки энтропии для специальной меры Бернулли", *Модел. и анализ информ. систем*, **24**:5 (2017), 521—536; [Timofeev E. A., "Existence of an unbiased entropy estimator for the special Bernoulli measure", *Modeling and Analysis of Information Systems*, **24**:5 (2017), 521—536, (in Russian).]

Timofeev E. A., "Existence of an Unbiased Consistent Entropy Estimator for the Special Bernoulli Measure", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 267–278.

DOI: 10.18255/1818-1015-2019-2-267-278

Abstract. Let $\Omega = \mathcal{A}^{\mathbb{N}}$ be a space of right-sided infinite sequences drawn from a finite alphabet $\mathcal{A} = \{0, 1\}$, $\mathbb{N} = \{1, 2, \dots\}$,

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{\infty} |x_k - y_k| 2^{-k}$$

– a metric on $\Omega = \mathcal{A}^{\mathbb{N}}$, and μ – a probability measure on Ω. Let $\boldsymbol{\xi_0}, \boldsymbol{\xi_1}, \dots, \boldsymbol{\xi_n}$ be independent identically distributed points on Ω. We study the estimator $\eta_n^{(k)}(\gamma)$ of the reciprocal of the entropy $1/h$ that are defined as

$$\eta_n^{(k)}(\gamma) = k \left(r_n^{(k)}(\gamma) - r_n^{(k+1)}(\gamma) \right),$$

where

$$r_n^{(k)}(\gamma) = \frac{1}{n+1} \sum_{j=0}^{n} \gamma \left(\min_{i:i \neq j}{}^{(k)} \rho(\boldsymbol{\xi_i}, \boldsymbol{\xi_j}) \right),$$

$\min^{(k)}\{X_1, \dots, X_N\} = X_k$, if $X_1 \leq X_2 \leq \dots \leq X_N$. Number k and a function $\gamma(t)$ are auxiliary parameters. The main result of this paper is

Theorem. Let μ be the Bernoulli measure with probabilities $p_0, p_1 > 0$, $p_0 + p_1 = 1$, $p_0 = p_1^2$, then $\forall \varepsilon > 0$ $\exists$ some continuous function $\gamma(t)$ such that

$$\left| \mathsf{E} \eta_n^{(k)}(\gamma) - \frac{1}{h} \right| < \varepsilon, \quad \mathsf{Var}\ \eta_n^{(k)}(\gamma) \to 0,\ n \to \infty.$$

Keywords: measure, metric, entropy, estimator, unbias, self-similar, Bernoulli measure

On the authors:
Timofeev Evgeniy Alexandrovich,orcid.org/0000-0002-3094-4390, ScD, professor
P.G. Demidov Yaroslavl State University,
Sovetskaya str., 14, Yaroslavl, 150003, Russia, e-mail: timofeevEA@gmail.com

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 279–296
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 279–296

Computing methodologies and applications

DOI: 10.18255/1818-1015-2019-2-279-296

УДК 514.17+517.51+519.6

Линейная интерполяция на евклидовом шаре в $\mathbb{R}^n$

Невский М. В., Ухалов А. Ю.

Поступила в редакцию 8 декабря 2018
После доработки 21 февраля 2019
Принята к публикации 25 февраля 2019

Аннотация. Пусть $x^{(0)} \in \mathbb{R}^n, R > 0$. Через $B = B(x^{(0)}; R)$ обозначим евклидов шар в $\mathbb{R}^n$, задаваемый неравенством $\|x - x^{(0)}\| \le R$, $\|x\| := \left(\sum_{i=1}^n x_i^2\right)^{1/2}$. Положим $B_n := B(0,1)$. Под $C(B)$ будем понимать пространство непрерывных функций $f : B \to \mathbb{R}$ с нормой $\|f\|_{C(B)} := \max_{x \in B} |f(x)|$, под $\Pi_1(\mathbb{R}^n)$ — совокупность многочленов от n переменных степени ≤ 1, то есть линейных функций на $\mathbb{R}^n$. Пусть $x^{(1)}, \ldots, x^{(n+1)}$ — вершины n-мерного невырожденного симплекса $S \subset B$. Интерполяционный проектор $P : C(B) \to \Pi_1(\mathbb{R}^n)$, соответствующий S, определяется равенствами $Pf\left(x^{(j)}\right) = f\left(x^{(j)}\right)$. Через $\|P\|_B$ обозначим норму P как оператора из $C(B)$ в $C(B)$. Определим $\theta_n(B)$ как минимальную величину $\|P\|_B$ при условии $x^{(j)} \in B$. В статье получена формула для вычисления $\|P\|_B$ через $x^{(0)}$, R и коэффициенты базисных многочленов Лагранжа, соответствующих S. Более подробно исследован случай, когда S — правильный симплекс, вписанный в B_n. Доказано, что в этой ситуации справедливо равенство $\|P\|_{B_n} = \max\{\psi(a), \psi(a+1)\}$, где $\psi(t) = \frac{2\sqrt{n}}{n+1}\big(t(n+1-t)\big)^{1/2} + \left|1 - \frac{2t}{n+1}\right|$ $(0 \le t \le n+1)$, целое a имеет вид $a = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$. Для такого проектора $\sqrt{n} \le \|P\|_{B_n} \le \sqrt{n+1}$, причём равенство $\|P\|_{B_n} = \sqrt{n+1}$ имеет место тогда и только тогда, когда число $\sqrt{n+1}$ является целым. Приводятся точные значения $\theta_n(B_n)$ для $1 \le n \le 4$. Даются результаты компьютерных вычислений, дополняющие теоретический анализ. Обсуждаются некоторые другие вопросы, связанные с интерполяцией на евклидовом шаре, в том числе открытые.

Ключевые слова: n-мерный симплекс, n-мерный шар, линейная интерполяция, проектор, норма

Для цитирования: Невский М. В., Ухалов А. Ю., "Линейная интерполяция на евклидовом шаре в $\mathbb{R}^n$", *Моделирование и анализ информационных систем*, **26**:2 (2019), 279–296.

Об авторах:
Невский Михаил Викторович, orcid.org/0000-0002-6392-7618, доктор физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: mnevsk55@yandex.ru

Ухалов Алексей Юрьевич, orcid.org/0000-0001-6551-5118, кандидат физ.-мат. наук,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: alex-uhalov@yandex.ru

1. Введение

Начнём с основных определений и обозначений. Всюду далее $n \in \mathbb{N}$. Элемент $x \in \mathbb{R}^n$ будем записывать в виде $x = (x_1, \ldots, x_n)$. Через e_1, …, e_n обозначим канонический

базис $\mathbb{R}^n$. Положим

$$\|x\| := \sqrt{(x,x)} = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2},$$

$$B\left(x^{(0)}; R\right) := \{x \in \mathbb{R}^n : \|x - x^{(0)}\| \le R\} \quad \left(x^{(0)} \in \mathbb{R}^n, R > 0\right),$$

$$B_n := B(0;1), \quad Q_n := [0,1]^n, \quad Q_n' := [-1,1]^n.$$

Всюду далее S есть невырожденный симплекс в $\mathbb{R}^n$. Через $c(S)$ обозначим центр тяжести S. Под τS будем понимать результат гомотетии S относительно точки $c(S)$ с коэффициентом τ. Под $\mathrm{ver}(G)$ понимается совокупность вершин выпуклого многогранника G. Для замкнутого ограниченного $\Omega \subset \mathbb{R}^n$ введём в рассмотрение величину $\xi(\Omega; S) := \min\{\sigma \ge 1 : \Omega \subset \sigma S\}$. Это число мы называем *коэффициентом поглощения (absorption index) множества Ω симплексом S.*

Обозначим вершины симплекса S через $x^{(j)} = \left(x_1^{(j)}, \ldots, x_n^{(j)}\right)$, $1 \le j \le n+1$. Матрица вершин

$$\mathbf{A} := \begin{pmatrix} x_1^{(1)} & \ldots & x_n^{(1)} & 1 \\ x_1^{(2)} & \ldots & x_n^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(n+1)} & \ldots & x_n^{(n+1)} & 1 \end{pmatrix}$$

является невырожденной. Положим $\mathbf{A}^{-1} = (l_{ij})$. Линейные многочлены $\lambda_j(x) = l_{1j}x_1 + \ldots + l_{nj}x_n + l_{n+1,j}$, коэффициенты которых составляют столбцы $\mathbf{A}^{-1}$, обладают свойством $\lambda_j\left(x^{(k)}\right) = \delta_j^k$. Мы называем λ_j *базисными многочленами Лагранжа, соответствующими S.* Числа $\lambda_j(x)$ являются барицентрическими координатами точки $x \in \mathbb{R}^n$ относительно S. Симплекс S задаётся системой линейных неравенств $\lambda_j(x) \ge 0$. Подробнее о многочленах λ_j см. [4, гл. 1].

Положим

$$\xi_n(\Omega) := \min\{\xi(\Omega; S) : \ S \text{ — } n\text{-мерный симплекс}, \mathrm{ver}(S) \subset \Omega, \ \mathrm{vol}(S) \ne 0\},$$

$\xi_n := \xi_n(Q_n)$. Разнообразные оценки чисел ξ_n получены в [2–5, 14] и других работах авторов. Здесь лишь отметим, что $n \le \xi_n < n+1$. Однако точные значения ξ_n пока известны лишь для $n = 2$, $n = 5$, $n = 9$ и бесконечной совокупности тех n, для каждого из которых существует матрица Адамара порядка $n+1$. Во всех этих ситуациях, кроме $n = 2$, выполняется $\xi_n = n$, в то время как $\xi_2 = \frac{3\sqrt{5}}{5} + 1 = 2.34\ldots$ До сих пор $n = 2$ остаётся единственным чётным n, для которого ξ_n вычислено точно. Число ξ_n для евклидова шара, т. е. точное значение $\xi_n(B_n)$, найдено в [7]. Именно для любого $S \subset B_n$ верно $\xi(B_n; S) \ge n$, причём равенство имеет место лишь для правильного симплекса S, вписанного в граничную сферу (для краткости в дальнейшем будем говорить, что такой симплекс вписан в шар). Таким образом, всегда $\xi_n(B_n) = n$.

Через $C(\Omega)$ обозначим пространство непрерывных функций $f : \Omega \to \mathbb{R}$ с равномерной нормой

$$\|f\|_{C(\Omega)} := \max_{x \in Q_n} |f(x)|.$$

Под $\Pi_1(\mathbb{R}^n)$ будем понимать совокупность многочленов от n переменных степени не выше 1 или линейных функций на $\mathbb{R}^n$.

Пусть $x^{(j)} \in \Omega$, $1 \le j \le n+1$, — вершины n-мерного невырожденного симплекса S. Будем говорить, что интерполяционный проектор $P : C(\Omega) \to \Pi_1(\mathbb{R}^n)$ соответствует симплексу S, если его узлы интерполяции совпадают с точками $x^{(j)}$. Этот проектор определяется равенствами $Pf\left(x^{(j)}\right) = f_j := f\left(x^{(j)}\right)$. Справедлив аналог интерполяционной формулы Лагранжа

$$Pf(x) = \sum_{j=1}^{n+1} f\left(x^{(j)}\right) \lambda_j(x). \tag{1}$$

Обозначим через $\|P\|_\Omega$ норму P как оператора из $C(\Omega)$ в $C(\Omega)$. Из (1) следует, что

$$\|P\|_\Omega = \sup_{\|f\|_{C(\Omega)}=1} \|Pf\|_{C(\Omega)} = \sup_{-1\le f_j\le 1} \max_{x\in\Omega} \sum_{j=1}^{n+1} f_j\lambda_j(x).$$

Выражение $\sum f_j\lambda_j(x)$ линейно по x и $f_1, \dots, f_{n+1}$, поэтому

$$\|P\|_\Omega = \max_{f_j=\pm 1} \max_{x\in\Omega} \sum_{j=1}^{n+1} f_j\lambda_j(x) = \max_{x\in\Omega} \max_{f_j=\pm 1} \sum_{j=1}^{n+1} f_j\lambda_j(x) = \max_{x\in\Omega} \sum_{j=1}^{n+1} |\lambda_j(x)|. \tag{2}$$

Если же Ω — выпуклый многогранник в $\mathbb{R}^n$ (например, $\Omega = Q_n$), то на этом пути получается и более простое равенство

$$\|P\|_\Omega = \max_{x\in \mathrm{ver}(\Omega)} \sum_{j=1}^{n+1} |\lambda_j(x)|.$$

Будем говорить, что точка $x \in \Omega$ является 1-*точкой* Ω *относительно* S, если для проектора $P : C(\Omega) \to \Pi_1(\mathbb{R}^n)$ с узлами в вершинах S выполняется равенство $\|P\| = \sum |\lambda_j(x)|$, причём среди чисел $\lambda_j(x)$ имеется ровно одно отрицательное. Для любого P и соответствующего S в [1, теорема 2.1] установлено соотношение

$$\frac{n+1}{2n}\Big(\|P\|_\Omega - 1\Big) + 1 \le \xi(\Omega; S) \le \frac{n+1}{2}\Big(\|P\|_\Omega - 1\Big) + 1. \tag{3}$$

Если в Ω существует 1-точка относительно S, то правое соотношение является равенством. (Последнее утверждение доказано в [1] в эквивалентном виде; понятие 1-вершины куба было введено позднее в [2].)

Через $\theta_n(\Omega)$ обозначим минимальную величину $\|P\|_\Omega$ при условии $x^{(j)} \in \Omega$. Для чисел $\theta_n := \theta_n(Q_n)$ в цикле работ первого автора доказана асимптотика $\theta_n \asymp \sqrt{n}$ (различные оценки систематизированы в [4]). В дальнейшем ряд оценок был улучшен М. В. Невским и А. Ю. Ухаловым, а также учениками второго автора (см., например, [5, 6]). На сегодняшний день точные значения θ_n известны лишь для $n = 1, 2, 3$ и 7. Именно, $\theta_1 = 1$, $\theta_2 = \frac{2\sqrt{5}}{5} + 1 = 1.89\dots$, $\theta_3 = 2$, $\theta_7 = \frac{5}{2}$. Для всех этих размерностей

$$\xi_n = \frac{n+1}{2}\Big(\theta_n - 1\Big) + 1.$$

В настоящей статье рассматривается случай $\Omega = B_n$. Поскольку для $S \subset B_n$ выполняется $\xi(B_n; S) \geq n$, то для соответствующего проектора P правое неравенство из (3) даёт

$$\|P\|_{B_n} \geq 3 - \frac{4}{n+1}. \tag{4}$$

Таким образом, всегда $\theta_n(B_n) \geq 3 - \frac{4}{n+1}$.

Заметим также, что если для интерполяционного проектора $P : C(B_n) \to \Pi_1(\mathbb{R}^n)$ в (4) выполняется равенство, то соответствующий симплекс S является правильным и вписан в B_n. Действительно, в этом случае

$$\frac{n+1}{2}\Big(\|P\|_{B_n} - 1\Big) + 1 = n,$$

и (3) даёт $\xi(B_n; S) \leq n$. Значит $\xi(B_n; S) = n$, а это выполняется лишь для правильного симплекса, вписанного в B_n. Поэтому из равенства $\theta_n(B_n) = 3 - \frac{4}{n+1}$ следует, что симплекс, соответствующий минимальному по норме проектору, обязательно имеет отмеченный вид.

Если S является правильным симплексом, вписанным в B_n, и существует 1-точка B_n относительно S, то соотношение (4) является равенством. Если же S не является правильным симплексом или не вписан в шар, то неравенство (4) является строгим. Как мы установим, 1-точка шара относительно вписанного правильного симплекса существует лишь при $1 \leq n \leq 4$. Начиная с $n = 5$ равенство (4) не выполняется ни при каких n и P, поэтому для $n \geq 5$ верно $\theta_n(B_n) > 3 - \frac{4}{n+1}$.

Настоящая статья содержит обоснование этого и других результатов для линейной интерполяции на шаре. В частности, мы получим точную формулу для нормы интерполяционного проектора, соответствующего вписанному правильному симплексу. Базируясь на этой формуле, мы покажем, что $\theta_n(B_n) \leq \sqrt{n+1}$, причём равенство здесь возможно лишь в случае $n = m^2 - 1$, $m \geq 2$.

2. Редукция в задаче о минимальном проекторе

Интерполяционный проектор $P : C(B_n) \to \Pi_1(\mathbb{R}^n)$ назовём *минимальным*, если для него $\|P\|_{B_n} = \theta_n(B_n)$. Существование минимального проектора вытекает из непрерывности $\|P\|$ как функции узлов, которую можно рассматривать на замкнутом ограниченном подмножестве пространства $\mathbb{R}^m$, $m = n(n+1)$, задаваемом ограничениями $x^{(j)} \in B_n$, $\det \mathbf{A} \geq \varepsilon_n > 0$. Покажем, что минимальный проектор может быть реализован по набору узлов, принадлежащих граничной сфере $\|x\| = 1$.

Пусть P — интерполяционный проектор с узлами $x^{(j)} \in B_n$, λ_j — базисные многочлены Лагранжа соответствующего симплекса S. Допустим, что не все $x^{(j)}$ принадлежат граничной сфере. Построим проектор P', для которого число узлов на границе B_n больше на 1, чем у исходного, и такой, что $\|P'\| \leq \|P\|$. Предположим, что узел $x^{(i)}$ проектора P лежит строго внутри B_n. Пусть z — ортогональная проекция $x^{(i)}$ на гиперплоскость $\lambda_i(x) = 0$, y — точка пересечения границы шара и прямой $(zx^{(i)})$, принадлежащая тому же полупространству, что и $x^{(i)}$. Тогда

$$0 < \sigma := \frac{\|y - x^{(i)}\|}{\|z - x^{(i)}\|} < 1.$$

Обозначим через T сжатие пространства $\mathbb{R}^n$ с коэффициентом σ относительно гиперплоскости $\lambda_i(x) = 0$ в ортогональном направлении. Рассмотрим проектор P_1 по тем же узлам, но определяемый через эллипсоид $T(B_n)$ — результат сжатия шара. Так как $x^{(i)}$ принадлежит границе $T(B_n)$, а остальные узлы лежат в гиперплоскости $\lambda_i(x) = 0$, то число узлов на границе $T(B_n)$ по сравнению с числом исходных узлов на границе B_n увеличится ровно на 1. Рассмотрим теперь обратное преобразование T^{-1}. Это невырожденное аффинное отображение переводит эллипсоид $T(B_n)$ обратно в шар B_n, а симплекс S — в некоторый симплекс $S' = T^{-1}(S)$. Очевидно, что i-я вершина S' совпадает с y, а другие вершины S' — те же, что и у S. Пусть P' — интерполяционный проектор, узлы которого находятся в вершинах S', вновь рассматриваемый на шаре B_n. Остаётся сравнить нормы проекторов. Поскольку $T(B_n) \subset B_n$, имеем

$$\|P_1\|_{T(B_n)} = \max_{x \in T(B_n)} \sum_{j=1}^{n+1} |\lambda_j(x)| \le \max_{x \in B_n} \sum_{j=1}^{n+1} |\lambda_j(x)| = \|P\|_{B_n}.$$

Так как пространства $C((B_n))$ и $C(B_n)$ изометричны, справедливо равенство $\|P'\|_{B_n} = \|P_1\|_{T(B_n)}$. Следовательно, $\|P'\|_{B_n} \le \|P\|_{B_n}$.

Применяя указанную процедуру нужное число раз можно переместить все узлы интерполяции на границу шара, при этом норма проектора не возрастёт. Значит, существует минимальный проектор, все узлы которого принадлежат граничной сфере. Это свойство, в частности, может применяться при численном решении задачи минимизации нормы проектора.

3. Норма проектора при интерполяции на шаре

Пусть $B = B(x^{(0)}; R)$, $P : C(B) \to \Pi_1(\mathbb{R}^n)$ — интерполяционный проектор по набору узлов $x^{(j)} \in B$, $\lambda_j(x) = l_{1j}x_1 + \ldots + l_{nj}x_n + l_{n+1,j}$ — базисные многочлены Лагранжа симплекса S, вершины которого совпадают с узлами интерполяции. Тогда, как отмечалось,

$$\|P\|_B = \max_{x \in B} \sum_{j=1}^{n+1} |\lambda_j(x)|, \tag{5}$$

см. (2) для $\Omega = B$. Дополним (5) другим выражением для нормы проектора.

Теорема 1. *Имеет место равенство*

$$\|P\|_B = \max_{f_j = \pm 1} \left[R \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2 \right)^{1/2} + \left| \sum_{j=1}^{n+1} f_j \left(\sum_{i=1}^{n} l_{ij} x_i^{(0)} + l_{n+1,j} \right) \right| \right] =$$

$$= \max_{f_j = \pm 1} \left[R \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2 \right)^{1/2} + \left| \sum_{j=1}^{n+1} f_j \lambda_j(x^{(0)}) \right| \right]. \tag{6}$$

Доказательство. В выражении для $\|P\|_B$ запишем повторный максимум в другом порядке, нежели при получении (5):

$$\|P\|_B = \max_{f_j=\pm1} \max_{x\in B} \sum_{j=1}^{n+1} f_j\lambda_j(x) = \max_{f_j=\pm1} \max_{x\in B} \left|\sum_{j=1}^{n+1} f_j\lambda_j(x)\right|. \tag{7}$$

При фиксированных f_j функция

$$\Lambda(x) := \sum_{j=1}^{n+1} f_j\lambda_j(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij}\right) x_i + \sum_{j=1}^{n+1} f_j l_{n+1,j}$$

является линейной по x. Максимум модуля $\Lambda(x)$ на шаре $B = B(x^{(0)}; R)$ достигается в одной из двух точек пересечения сферы $\|x-x^{(0)}\| = R$ и прямой, проходящей через центр шара $x^{(0)}$ и имеющей направляющий вектор $v = (v_1, \ldots, v_n)$, где $v_i = \sum\limits_{j=1}^{n+1} f_j l_{ij}$. Эти точки имеют вид

$$x_+ = x^{(0)} + \frac{R}{\|v\|}v, \quad x_- = x^{(0)} - \frac{R}{\|v\|}v.$$

Функция $L(x) := \Lambda(x) - \Lambda(0)$ является аддитивной и однородной, поэтому

$$L(x_+) = L\left(x^{(0)} + \frac{R}{\|v\|}v\right) = L\left(x^{(0)}\right) + \frac{R}{\|v\|}L(v),$$

$$\Lambda(x_+) = L(x_+) + \Lambda(0) = \Lambda\left(x^{(0)}\right) + \frac{R}{\|v\|}\Big[\Lambda(v) - \Lambda(0)\Big].$$

Заметим, что

$$\Lambda(v) - \Lambda(0) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij}\right) v_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij}\right)^2 = \|v\|^2,$$

поэтому $\Lambda(x_+) = \Lambda\left(x^{(0)}\right) + R\|v\|$. Аналогично $\Lambda(x_-) = \Lambda\left(x^{(0)}\right) - R\|v\|$. Тем самым,

$$\max_{x\in B} |\Lambda(x)| = \max(|\Lambda(x_+)|, |\Lambda(x_-)|) = R\|v\| + \left|\Lambda\left(x^{(0)}\right)\right|.$$

Равенство (7) теперь даёт

$$\|P\|_B = \max_{f_j=\pm1} \max_{x\in B} |\Lambda(x)| = \max_{f_j=\pm1} \left[R\|v\| + \left|\Lambda\left(x^{(0)}\right)\right|\right].$$

Остаётся учесть, что

$$\|v\| = \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij}\right)^2\right)^{1/2},$$

$$\Lambda\left(x^{(0)}\right) = \sum_{j=1}^{n+1} f_j\lambda_j(x^{(0)}) = \sum_{j=1}^{n+1} f_j \left(\sum_{i=1}^{n} l_{ij}x_i^{(0)} + l_{n+1,j}\right).$$

Теорема доказана. □

В случае, когда центр тяжести симплекса совпадает с центром шара, формула (6) заметно упрощается.

Следствие 1. *Пусть* $c(S) = x^{(0)}$, *тогда*

$$\|P\|_B = \max_{f_j=\pm 1} \left[R \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2 \right)^{1/2} + \frac{1}{n+1} \left| \sum_{j=1}^{n+1} f_j \right| \right]. \tag{8}$$

Доказательство. Числа $\lambda_j(x^{(0)})$ суть барицентрические координаты точки $x^{(0)}$ относительно симплекса S. В силу равенств

$$x^{(0)} = c(S) = \frac{1}{n+1} \sum_{j=1}^{n+1} x^{(j)}$$

имеем $\lambda_j(x^{(0)}) = \frac{1}{n+1}$ при любом j. Поэтому (8) получается из (6). □

4. Норма проектора для правильного симплекса, вписанного в шар

В этом пункте S — правильный симплекс, вписанный в n-мерный шар $B = (x^{(0)}; R)$, $P : C(B) \to \Pi_1(\mathbb{R}^n)$ — соответствующий интерполяционный проектор. Очевидно, $\|P\|_B$ не зависит ни от центра $x^{(0)}$ и радиуса R шара, ни от выбора правильного симплекса, вписанного в этот шар. Иначе говоря, $\|P\|_B$ есть функция размерности n. В этом пункте мы получим явный вид этой функции и установим её оценки.

Для $0 \le t \le n+1$ введём в рассмотрение функцию

$$\psi(t) := \frac{2\sqrt{n}}{n+1} \Big(t(n+1-t) \Big)^{1/2} + \left| 1 - \frac{2t}{n+1} \right|. \tag{9}$$

Обозначим $a := \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$, где $\lfloor s \rfloor$ — целая часть s.

Теорема 2. *Справедливы соотношения*

$$\|P\|_B = \max\{\psi(a), \psi(a+1)\}, \tag{10}$$

$$\sqrt{n} \le \|P\|_B \le \sqrt{n+1}. \tag{11}$$

При этом $\|P\|_B = \sqrt{n}$ *лишь в случае* $n = 1$, *а равенство* $\|P\|_B = \sqrt{n+1}$ *выполняется тогда и только тогда, когда* $\sqrt{n+1}$ — *целое число.*

Доказательство. Сначала докажем (10). Если $n = 1$, то $\psi(t) = \sqrt{t(2-t)} + |1-t|$, $a = 0$, $\psi(a) = \psi(a+1) = 1$. Так как $\|P\|_B = 1$, то равенство (10) верно.

Пусть $n \ge 2$. Рассмотрим симплекс S с вершинами

$$x^{(1)} = e_1, \quad \ldots, \quad x^{(n)} = e_n, \quad x^{(n+1)} = \left(\frac{1-\sqrt{n+1}}{n}, \ldots, \frac{1-\sqrt{n+1}}{n} \right).$$

Этот симплекс является правильным, так как длина любого его ребра равна $\sqrt{2}$. Симплекс S вписан в шар $B = B(x^{(0)}; R)$, где

$$x^{(0)} = c(S) = \left(\frac{1 - \sqrt{\frac{1}{n+1}}}{n}, \ldots, \frac{1 - \sqrt{\frac{1}{n+1}}}{n} \right), \quad R = \sqrt{\frac{n}{n+1}}.$$

Заметим, что $(n+1)$-я вершина S получается перемещением нулевой вершины координатного симплекса $x_i \geq 0$, $\sum x_i \leq 1$ в сторону от гиперплоскости $\sum x_i = 1$. Для нас важно, что S переходит в себя при любом переобозначении координат. В силу замечания, сделанного в начале пункта, достаточно найти $\|P\|_B$ именно для этого симплекса S.

Матрицы $\mathbf{A}$ и $\mathbf{A}^{-1}$, соответствующие S, имеют вид

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 1 \\ -\tau & -\tau & \ldots & -\tau & 1 \end{pmatrix}, \quad \mathbf{A}^{-1} = \frac{1}{\sqrt{n+1}} \begin{pmatrix} \sigma & -\tau & \ldots & -\tau & -1 \\ -\tau & \sigma & \ldots & -\tau & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\tau & -\tau & \ldots & \sigma & -1 \\ \tau & \tau & \ldots & \tau & 1 \end{pmatrix}. \quad (12)$$

Мы положили

$$\sigma := \frac{(n-1)\sqrt{n+1} + 1}{n}, \quad \tau := \frac{\sqrt{n+1} - 1}{n}. \quad (13)$$

Поскольку $c(S) = x^{(0)}$, для вычисления $\|P\|_B$ воспользуемся формулой (8), в которой возьмём $R = \sqrt{\frac{n}{n+1}}$:

$$\|P\|_B = \max_{f_j = \pm 1} \left[\sqrt{\frac{n}{n+1}} \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2 \right)^{1/2} + \frac{1}{n+1} \left| \sum_{j=1}^{n+1} f_j \right| \right].$$

Здесь l_{ij} — элементы матрицы $\mathbf{A}^{-1}$, см. (12).

Обозначим через k число f_j, равных -1. Тогда число f_j, равных 1, есть $n+1-k$. Так как S не меняется при переобозначении координат, будем считать, что $f_1 = \ldots = f_k = -1$, $f_{k+1} = \ldots = f_{n+1} = 1$. В силу того, что максимизируемая величина не меняется при перемене знака сразу у всех f_j, можно ограничиться интервалом $1 \leq k \leq \frac{n+1}{2}$. Итак,

$$\|P\|_B = \max_{1 \leq k \leq \frac{n+1}{2}} \left[\sqrt{\frac{n}{n+1}} \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2 \right)^{1/2} + \frac{n+1-2k}{n+1} \right], \quad (14)$$

причём здесь $f_j = -1$ для $1 \leq j \leq k$ и $f_j = 1$ для остальных j. Число $n+1-2k$ равно сумме, стоящей под знаком модуля. Учитывая множитель $\frac{1}{\sqrt{n+1}}$, стоящий в равенстве для $\mathbf{A}^{-1}$, преобразуем величину

$$W := (n+1) \sum_{i=1}^{n} \left(\sum_{j=1}^{n+1} f_j l_{ij} \right)^2.$$

Чтобы использовать явный вид l_{ij}, представим эту сумму в виде

$$W = (n+1)\sum_{i=1}^{k}\left(\sum_{j=1}^{n+1} f_j l_{ij}\right)^2 + (n+1)\sum_{i=k+1}^{n}\left(\sum_{j=1}^{n+1} f_j l_{ij}\right)^2 = W_1 + W_2.$$

Применение (12) и указанного распределения значений f_j даёт

$$W_1 = \sum_{i=1}^{k}(-\sigma + (k-1)\tau - (n-k)\tau - 1)^2 = k(2k\tau - \alpha)^2,$$

$$W_2 = \sum_{i=k+1}^{n}(k\tau + \sigma - (n-1-k)\tau - 1)^2 = (n-k)(2k\tau + \beta)^2,$$

где $\alpha = \sigma + (n+1)\tau + 1$, $\beta = \sigma - (n-1)\tau - 1$. Из (13) следует, что $\alpha = 2\sqrt{n+1}$, $\beta = 0$, поэтому

$$W = 4k(k\tau - \sqrt{n+1})^2 + (n-k)\cdot 4k^2\tau^2 = k^2(-8\sqrt{n+1}\tau + 4n\tau^2) + 4k(n+1) =$$

$$= -4k^2 + 4k(n+1) = 4k(n+1-k).$$

Таким образом,

$$\|P\|_B = \max_{1\le k\le \frac{n+1}{2}}\left[\sqrt{\frac{n}{n+1}}\left(\frac{1}{n+1}W\right)^{1/2} + \frac{n+1-2k}{n+1}\right] =$$

$$= \max_{1\le k\le \frac{n+1}{2}}\left[\frac{2\sqrt{n}}{n+1}\big(k(n+1-k)\big)^{1/2} + 1 - \frac{2k}{n+1}\right].$$

Принимая во внимание (9), приходим к равенству

$$\|P\|_B = \max_{1\le k\le \frac{n+1}{2}} \psi(k). \tag{15}$$

Осталось показать, что последний максимум совпадает с наибольшим из чисел $\psi(a)$ и $\psi(a+1)$, где $a = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$. Для этого проанализируем поведение $\psi(t)$ на всём промежутке $[0, n+1]$.

Функция

$$\psi(t) = \frac{2\sqrt{n}}{n+1}\Big(t(n+1-t)\Big)^{1/2} + \left|1 - \frac{2t}{n+1}\right|, \quad 0 \le t \le n+1$$

обладает свойствами $\psi(t) > 0$, $\psi(0) = \psi(n+1) = 1$, $\psi\left(\frac{n+1}{2}\right) = \sqrt{n}$, $\psi(n+1-t) = \psi(t)$. График $\psi(t)$ симметричен относительно прямой $t = \frac{n+1}{2}$. На каждой из половин отрезка $[0, n+1]$ функция $\psi(t)$ является вогнутой как сумма двух вогнутых функций. Действительно, для $0 \le t \le \frac{n+1}{2}$

$$\psi(t) = \varphi_1(t) + \varphi_2(t), \quad \varphi_1(t) := \frac{2\sqrt{n}}{n+1}\Big(t(n+1-t)\Big)^{1/2}, \quad \varphi_2(t) := 1 - \frac{2t}{n+1},$$

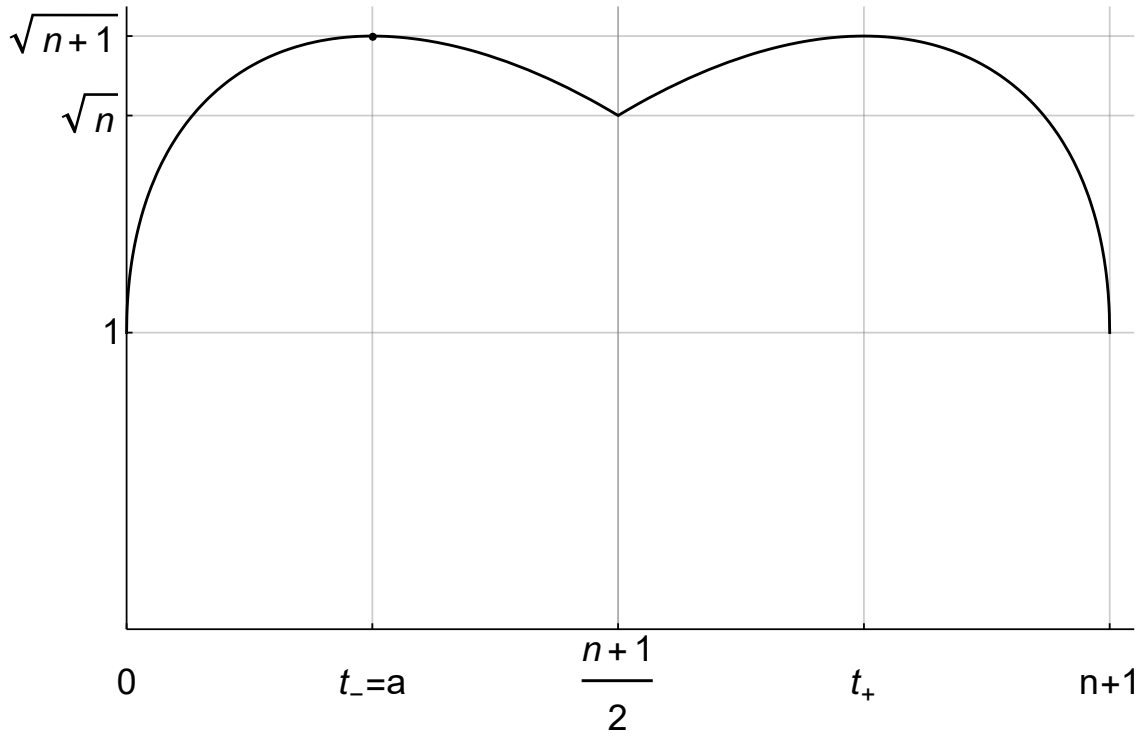


Рис. 1: График $\psi(t)$ для $n = 3$. Здесь $n + 1 = 4$, $t_- = a = 1$
Fig. 1: Graph of $\psi(t)$ for $n = 3$. Here $n + 1 = 4$, $t_- = a = 1$

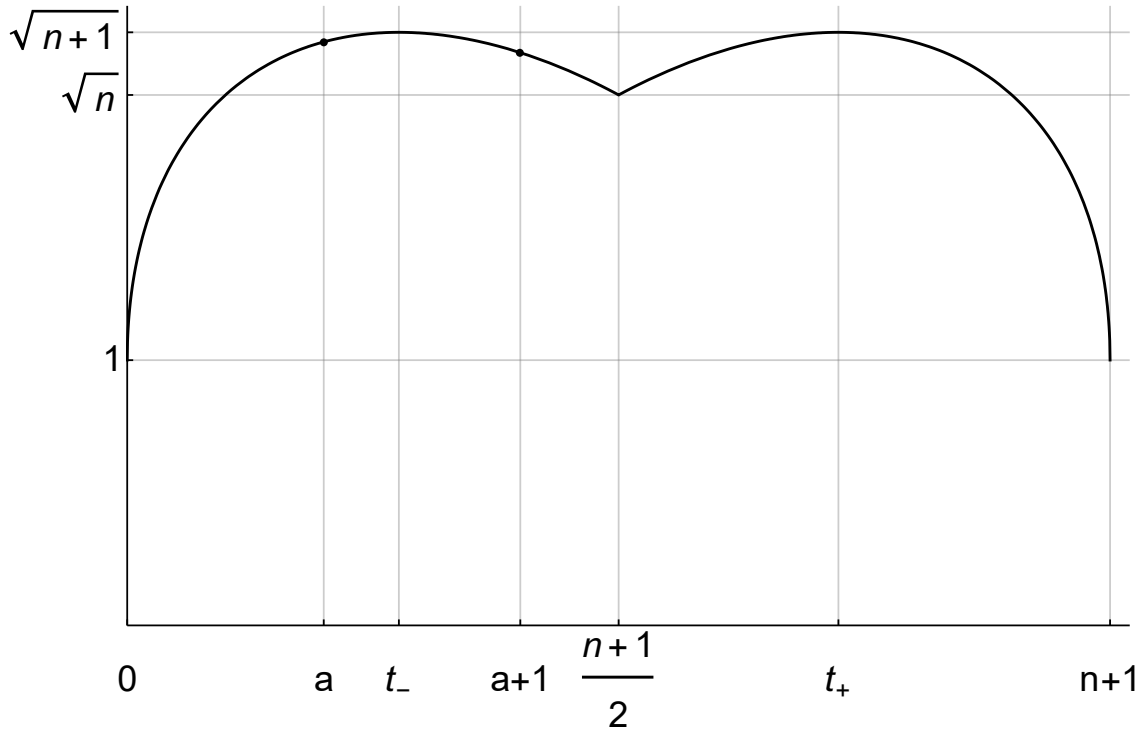


Рис. 2: График $\psi(t)$ для $n = 4$. Здесь $n + 1 = 5$, $a = 1$, $t_- = \frac{5-\sqrt{5}}{2}$
Fig. 2: Graph of $\psi(t)$ for $n = 4$. Here $n + 1 = 5$, $a = 1$, $t_- = \frac{5-\sqrt{5}}{2}$

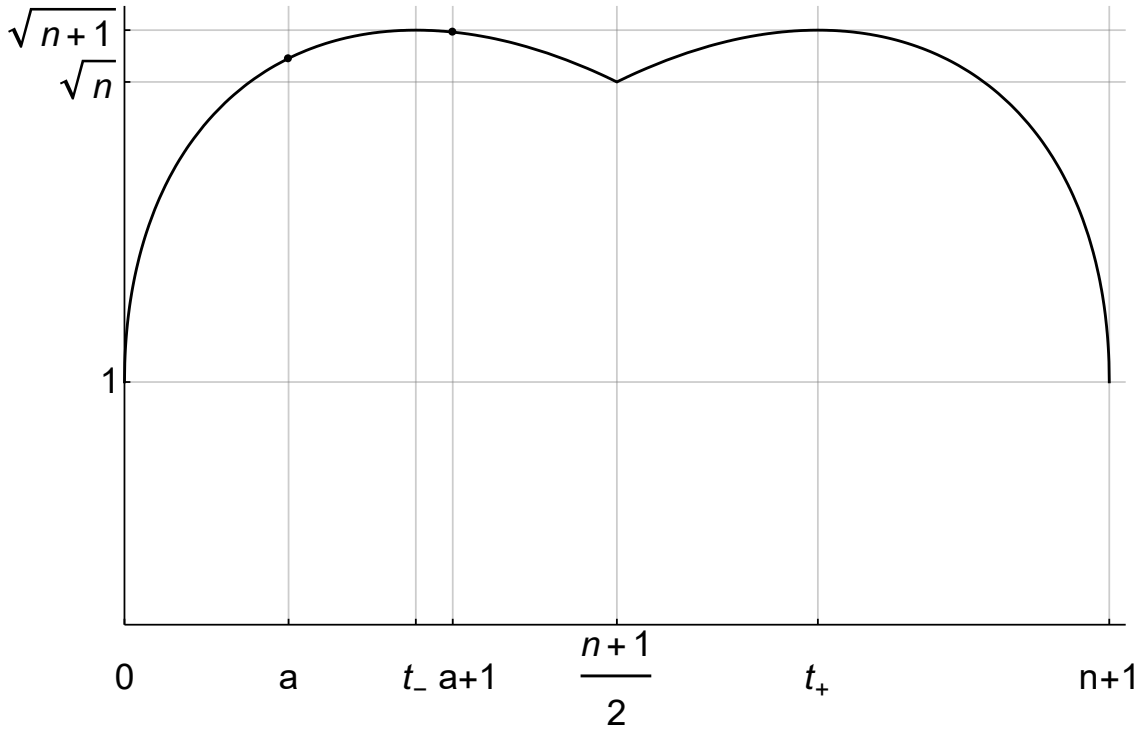


Рис. 3: График $\psi(t)$ для $n = 5$. Здесь $n + 1 = 5$, $a = 1$, $t_- = \frac{6-\sqrt{6}}{2}$
Fig. 3: Graph of $\psi(t)$ for $n = 5$. Here $n + 1 = 5$, $a = 1$, $t_- = \frac{6-\sqrt{6}}{2}$

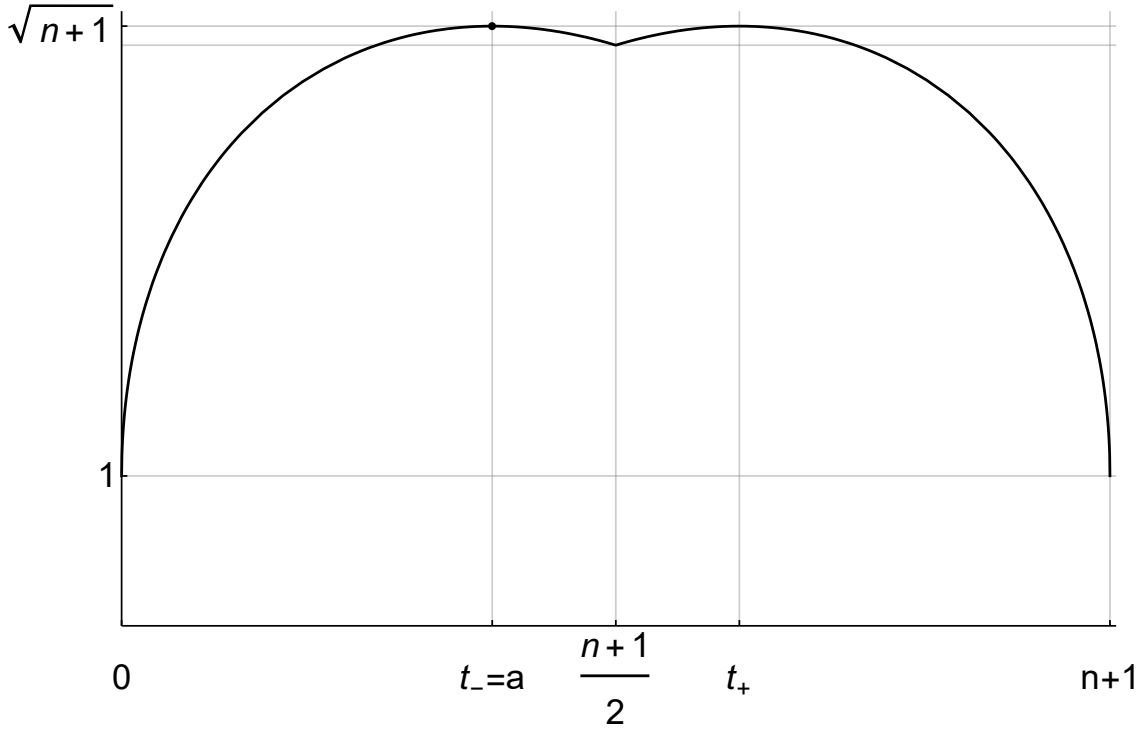


Рис. 4: График $\psi(t)$ для $n = 15$. Здесь $n + 1 = 16$, $t_- = a = 6$
Fig. 4: Graph of $\psi(t)$ for $n = 15$. Here $n + 1 = 16$, $t_- = a = 6$

причём $\varphi_1(t)$ вогнута как суперпозиция вогнутой функции $t(n+1-t)$ и возрастающей вогнутой функции $\sqrt{t}$, а $\varphi_2(t)$ является линейной. Производная $\psi'(t)$ обращается в нуль только в двух точках

$$t_- := \frac{n+1}{2} - \frac{\sqrt{n+1}}{2}, \quad t_+ := \frac{n+1}{2} + \frac{\sqrt{n+1}}{2},$$

внутренних соответственно для $\left[0, \frac{n+1}{2}\right]$ и $\left[\frac{n+1}{2}, n+1\right]$. Из вогнутости $\psi(t)$ на каждом из этих отрезков следует, что

$$\max_{0\le\psi(t)\le n+1} \psi(t) = \psi(t_-) = \psi(t_+) = \sqrt{n+1}.$$

При этом t_- — единственная точка максимума $\psi(t)$ на левой, а t_+ — на правой половине отрезка $[0, n+1]$. Значит, $\psi(t)$ возрастает при $0 \le t \le t_-$ и убывает при $t_- \le t \le \frac{n+1}{2}$. На левой половине $\psi(t)$ ведёт себя симметрично: возрастает при $\frac{n+1}{2} \le t \le t_+$ и убывает при $t_+ \le t \le n+1$.

Теперь ограничимся отрезком $\left[0, \frac{n+1}{2}\right]$. Так как $a = \lfloor t_- \rfloor$, то всегда $a \le t_- < a+1$. Пусть k — целое, $1 \le k \le \frac{n+1}{2}$. Из свойств $\psi(t)$ следует, что если $k < a$, то $\psi(k) < \psi(a)$, а если $k > a+1$, то $\psi(k) < \psi(a+1)$. С учётом (15) имеем

$$\|P\|_B = \max_{1\le k\le \frac{n+1}{2}} \psi(k) = \max\{\psi(a), \psi(a+1)\}.$$

Равенство (10) установлено.

Перейдём к неравенствам (11). Правое из них мы уже получили:

$$\|P\|_B = \max_{k\in\mathbb{Z}:\, 1\le k\le \frac{n+1}{2}} \psi(k) \le \max_{1\le t\le \frac{n+1}{2}} \psi(t) = \psi(t_-) = \sqrt{n+1}.$$

Опишем размерности n, для которых $\|P\|_B = \sqrt{n+1}$. Они характеризуются тем, что нестрогое неравенство в последнем соотношении является равенством, т. е. число t_- является целым.

Заметим, что каждая из точек t_- и t_+ является целочисленной тогда и только тогда, когда $\sqrt{n+1}$ есть целое число. Пусть, например,

$$t_- := \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} = d \in \mathbb{Z}.$$

Тогда $\sqrt{n+1} = n+1-2d$ — целое. Наоборот, если $\sqrt{n+1} = m \in \mathbb{Z}$, то $m(m-1)$ есть чётное число и $t_- = \frac{m(m-1)}{2}$ — целое. Утверждение для t_+ доказывается аналогично, а также выводится из предыдущего, поскольку $t_+ - t_- = \sqrt{n+1}$.

Мы получили, что для любой размерности вида $n = m^2 - 1$, m — целое, и только в этих ситуациях число t_- является целым и, значит,

$$\|P\|_B = \max_{k\in\mathbb{Z}:\, 1\le k\le \frac{n+1}{2}} \psi(k) = \max_{1\le t\le \frac{n+1}{2}} \psi(t) = \psi(t_-) = \sqrt{n+1} = m.$$

Эти равенства эквивалентны (10), поскольку в рассматриваемых случаях $a = \lfloor t_- \rfloor = t_-$ и $\|P\|_B = \psi(a)$. Для каждого из остальных n (т. е. отличных от чисел $m^2 - 1$) выполняется $\|P\| < \sqrt{n+1}$. Действительно, тогда $a < t_- < a+1$ и максимум в (15)

достигается либо при $k = a$, либо при $k = a + 1$. Для любого $n \neq m^2 - 1$ норма P, т. е. максимум $\psi(k)$ по целым k из $\left[1, \frac{n+1}{2}\right]$ строго меньше, чем максимум $\psi(t)$ по всему этому промежутку.

Осталось показать, что всегда $\|P\|_B \geq \sqrt{n}$, причём для $n > 1$ это неравенство является строгим. Если $n > 3$, то $\sqrt{n+1} > 2$, поэтому

$$a + 1 = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor + 1 \leq \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} + 1 < \frac{n+1}{2}.$$

Из (10) и свойств $\psi(t)$ имеем

$$\|P\|_B = \max\{\psi(a), \psi(a+1)\} \geq \psi(a+1) > \psi\left(\frac{n+1}{2}\right) = \sqrt{n} \qquad (n > 3).$$

Для $n = 2$ и $n = 3$ также верно $\|P\|_B > \sqrt{n}$. Поэтому норма P равна $\sqrt{n}$ лишь в случае $n = 1$. Теорема полностью доказана. □

Будем говорить, что натуральное m есть число Адамара, если существует матрица Адамара порядка m (об этих матрицах см., например, [10–13]). В случае, когда $n+1$ — число Адамара, точная по порядку нижняя оценка $\|P\|_B \geq c\sqrt{n}$ следует из предыдущих результатов первого автора. В 2006 г. им было установлено, что при любом n

$$\theta_n(Q) \geq \chi_n^{-1}\left(\frac{1}{\nu_n}\right) > \frac{1}{e}\sqrt{n-1}. \tag{16}$$

Здесь Q — n-мерный куб, χ_n — нормализованный многочлен Лежандра степени n, ν_n — максимальный объём симплекса, содержащегося в единичном кубе $Q_n = [0,1]^n$. По поводу свойств χ_n см. [8, 9]. Доказательство соотношений (16) приведено в [4]. Предположим дополнительно, что $n+1$ — число Адамара. Тогда существует правильный симплекс, вершины которого находятся в вершинах куба (см. [13]). Выберем в качестве Q куб, вписанный в евклидов шар B. Тогда указанный правильный симплекс, вписанный в куб, будет вписан также и в шар. Так как $Q \subset B$, то для соответствующего проектора P выполняются соотношения

$$\|P\|_B \geq \|P\|_Q \geq \theta_n(Q) \geq \chi_n^{-1}\left(\frac{1}{\nu_n}\right) > \frac{1}{e}\sqrt{n-1}.$$

5. Точные значения $\theta_n(B_n)$ для $1 \leq n \leq 4$

С помощью теоремы 2 и свойств, указанных во введении, для $n = 1, 2, 3, 4$ можно получить точные значения минимальной нормы проектора на n-мерном шаре и описать минимальные проекторы. Разумеется, $\theta_1(B_1) = 1$, но и этот случай укладывается в общую схему.

Теорема 3. *Имеют место равенства*

$$\theta_1(B_1) = 1, \quad \theta_2(B_2) = \frac{5}{3}, \quad \theta_3(B_3) = 2, \quad \theta_4(B_4) = \frac{11}{5}. \tag{17}$$

При $1 \leq n \leq 4$ равенство $\|P\|_{B_n} = \theta_n(B_n)$ выполняется лишь для проекторов, каждый из которых соответствует правильному симплексу, вписанному в B_n.

Доказательство. Вычислим норму проектора $P : C(B_n) \to \Pi_1(\mathbb{R}^n)$, соответствующего правильному симплексу, вписанному в B_n. По теореме 2

$$\|P\|_{B_n} = \max\{\psi(a), \psi(a+1)\},$$

где $\psi(t)$ определяется равенством (9) и $a = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$. Для $n = 1$ имеем $\psi(t) = \sqrt{t(2-t)} + |1-t|$, $a = 0$, $\psi(a) = \psi(a+1) = 1$ и $\|P\|_{B_1} = 1$. В случае $n = 2$ верно $\psi(t) = \frac{2\sqrt{2}}{3}\sqrt{t(3-t)} + \left|1 - \frac{2t}{3}\right|$, $a = 0$, $\psi(a) = 1$, $\psi(a+1) = \frac{5}{3}$ и $\|P\|_{B_2} = \psi(a+1) = \frac{5}{3}$. Если $n = 3$, то $\psi(t) = \frac{\sqrt{3}}{2}\sqrt{t(4-t)} + \left|1 - \frac{t}{2}\right|$, $a = 1$, $\psi(a) = \psi(a+1) = 2$, поэтому $\|P\|_{B_3} = 2$. Наконец, для $n = 4$ надо взять $\psi(t) = \frac{4}{5}\sqrt{t(5-t)} + \left|1 - \frac{2t}{5}\right|$, $a = 1$, $\psi(a) = \frac{11}{5}$, $\psi(a+1) = \frac{1}{5}\left(1 + 4\sqrt{6}\right)$, что даёт $\|P\|_{B_4} = \psi(a) = \frac{11}{5}$.

Теперь заметим, что для каждого $n = 1, 2, 3, 4$ выполняется равенство

$$\|P\|_{B_n} = 3 - \frac{4}{n+1}.$$

В силу оценки $\theta_n(B_n) \geq 3 - \frac{4}{n+1}$, справедливой вообще для любой размерности, при $1 \leq n \leq 4$ верно

$$\theta_n(B_n) = 3 - \frac{4}{n+1}. \tag{18}$$

Это совпадает с (17). Как отмечалось выше, в случае выполнения (18) любой проектор, норма которого минимальна, соответствует правильному симплексу, вписанному в B_n. Это завершает доказательство. □

6. Заключительные замечания

Из неравенства (11) теоремы 2 следует, что величина $\theta_n(B_n)$ минимальной нормы проектора при линейной интерполяции на единичном шаре $\|x\| \leq 1$ удовлетворяет неравенству

$$\theta_n(B_n) \leq \sqrt{n+1}. \tag{19}$$

По крайней мере для тех n, когда число $\sqrt{n+1}$ не является целым, т. е. $n \neq m^2 - 1$, это неравенство является строгим. Если же минимальную норму имеет проектор, узлы которого находятся в вершинах правильного симплекса, вписанного в сферу $\|x\| = 1$, то для этого n в предыдущих обозначениях

$$\theta_n(B_n) = \max\{\psi(a), \psi(a+1)\}. \tag{20}$$

Равенство (20) выполняется, например, при $1 \leq n \leq 4$, когда оно эквивалентно соотношению (18), см. раздел 5. Вопросы о точности оценки (19) по порядку n и о выполнении (20) при $n > 4$ предполагается осветить в другой статье.

В заключение приведём некоторые иллюстрации, результаты численного анализа и некоторые комментарии к ним. Графики функции

$$\psi(t) = \frac{2\sqrt{n}}{n+1}\Big(t(n+1-t)\Big)^{1/2} + \left|1 - \frac{2t}{n+1}\right|, \quad 0 \leq t \leq n+1,$$

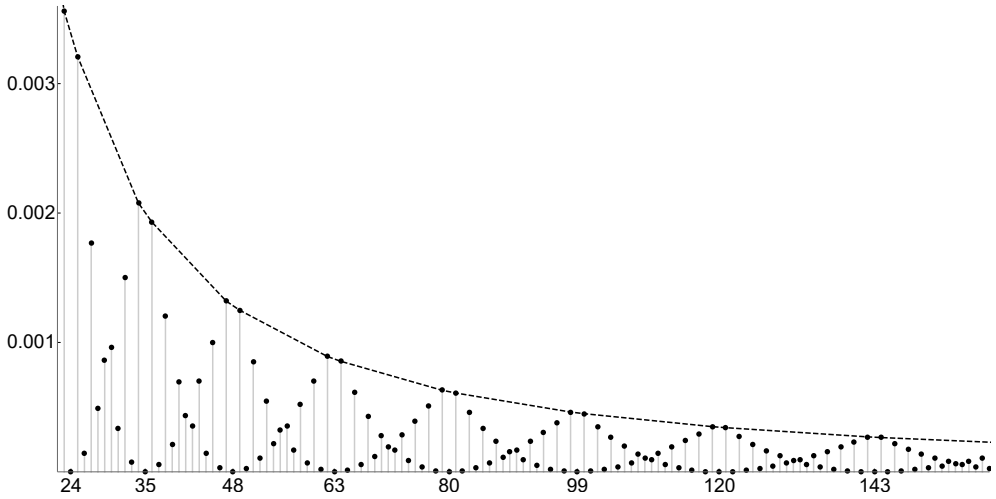


Рис. 5: Числа $d_n = \sqrt{n+1} - \|P\|_{B_n}$ для $23 \le n \le 160$
Fig. 5: The numbers $d_n = \sqrt{n+1} - \|P\|_{B_n}$ for $23 \le n \le 160$

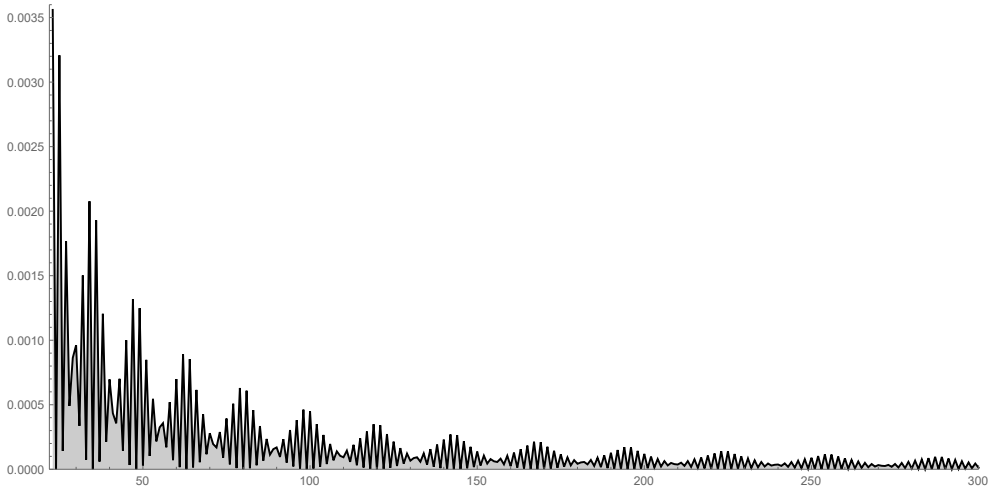


Рис. 6: Числа $d_n = \sqrt{n+1} - \|P\|_{B_n}$ для $23 \le n \le 300$
Fig. 6: The numbers $d_n = \sqrt{n+1} - \|P\|_{B_n}$ for $23 \le n \le 300$

для $n = 3, 4, 5, 15$ изображены на рис.1–4. Отмечены точки максимума этой функции $t_- := \frac{n+1}{2} - \frac{\sqrt{n+1}}{2}$, $t_+ := \frac{n+1}{2} + \frac{\sqrt{n+1}}{2}$, а также точки $a = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$ и $a+1$, одна из которых максимизирует $\psi(k)$ при целом $1 \le k \le \frac{n+1}{2}$. На рис. 5–6 для $n \ge 23$ представлены значения разности $d_n = \sqrt{n+1} - \|P\|_{B_n}$, где P — интерполяционный проектор, соответствующий правильному симплексу, вписанному в B_n. Как мы установили, $\|P\|_{B_n} \le \sqrt{n+1}$, причём равенство имеет место лишь для $n = 3, 8, 15, 24, 35, 48, 63, 80, \ldots$, т. е. размерностей вида $m^2 - 1$. Именно в этих точках $d_n = 0$, что и видно на рис. 5. Пунктирной линией обозначен график ломаной (линейного интерполяционного сплайна) l, построенной по узлам $n = m^2 - 2$, $n = m^2$ и значениям d_n в этих узлах. Всегда $d_n \le l(n)$, причём равенство достигается лишь для $n = m^2 - 2$ и $n = m^2$. При $n \to \infty$ монотонно $l(n) \to 0$. Таким образом, справедлива двусторонняя оценка $\sqrt{n+1} - l(n) \le \|P\|_{B_n} \le \sqrt{n+1}$. Как правое,

так и левое соотношения обращаются в равенства для бесконечного множества n. Эта оценка точнее, чем (2).

Таблица 1: Норма P для правильного симплекса, вписанного в B_n
Table 1: Norm of P for a regular simplex inscribed into B_n

n	t_-	a	$a+1$	$\psi(a)$	$\psi(a+1)$	k^*	$\|P\|_{B_n}$	$\|P\|_{B_n}$
1	$1-\frac{1}{\sqrt{2}}$	0	1	1	1	1	1	1
2	$\frac{3-\sqrt{3}}{2}$	0	1	1	$\frac{5}{3}$	1	$\frac{5}{3}$	1.6666 …
3	1	1	2	2	$\sqrt{3}$	1	2	2
4	$\frac{5-\sqrt{5}}{2}$	1	2	$\frac{11}{5}$	$\frac{1+4\sqrt{6}}{5}$	1	$\frac{11}{5}$	2.2
5	$3-\sqrt{\frac{3}{2}}$	1	2	$\frac{7}{3}$	$\frac{1+2\sqrt{10}}{3}$	2	$\frac{1+2\sqrt{10}}{3}$	2.4415 …
6	$\frac{7-\sqrt{7}}{2}$	2	3	$\frac{3+4\sqrt{15}}{7}$	$\frac{1+12\sqrt{2}}{7}$	2	$\frac{3+4\sqrt{15}}{7}$	2.6417 …
7	$4-\sqrt{2}$	2	3	$\frac{1+\sqrt{21}}{2}$	$\frac{1+\sqrt{105}}{4}$	3	$\frac{1+\sqrt{105}}{4}$	2.8117 …
8	3	3	4	3	$\frac{1+8\sqrt{10}}{9}$	3	3	3
9	$5-\sqrt{\frac{5}{2}}$	3	4	$\frac{2+3\sqrt{21}}{5}$	$\frac{1+6\sqrt{6}}{5}$	3	$\frac{2+3\sqrt{21}}{5}$	3.1495 …
10	$\frac{11-\sqrt{11}}{2}$	3	4	$\frac{5+8\sqrt{15}}{11}$	$\frac{3+4\sqrt{70}}{11}$	4	$\frac{3+4\sqrt{70}}{11}$	3.3151 …
11	$6-\sqrt{3}$	4	5	$\frac{1+2\sqrt{22}}{3}$	$\frac{1+\sqrt{385}}{6}$	4	$\frac{1+2\sqrt{22}}{3}$	3.4602 …
12	$\frac{13-\sqrt{13}}{2}$	4	5	$\frac{5+24\sqrt{3}}{13}$	$\frac{3+8\sqrt{30}}{13}$	5	$\frac{3+8\sqrt{30}}{13}$	3.6013 …
13	$7-\sqrt{\frac{7}{2}}$	5	6	$\frac{2+3\sqrt{65}}{7}$	$\frac{1+4\sqrt{39}}{7}$	5	$\frac{2+3\sqrt{65}}{7}$	3.7409 …
14	$\frac{15-\sqrt{15}}{2}$	5	6	$\frac{1+4\sqrt{7}}{3}$	$\frac{1+4\sqrt{21}}{5}$	6	$\frac{1+4\sqrt{21}}{5}$	3.8660 …
15	6	6	7	4	$\frac{1+3\sqrt{105}}{8}$	6	4	4
50	$\frac{51-\sqrt{51}}{2}$	21	22	$\frac{3+20\sqrt{35}}{17}$	$\frac{7+20\sqrt{319}}{51}$	22	$\frac{7+20\sqrt{319}}{51}$	7.1414 …
100	$\frac{101-\sqrt{101}}{2}$	45	46	$\frac{11+120\sqrt{70}}{101}$	$\frac{9+20\sqrt{2530}}{101}$	45	$\frac{11+120\sqrt{70}}{101}$	10.0494 …
1000	$\frac{1001-\sqrt{1001}}{2}$	484	485	$\frac{3+40\sqrt{5170}}{91}$	$\frac{31+200\sqrt{25026}}{1001}$	485	$\frac{31+200\sqrt{25026}}{1001}$	31.6385 …

В таблице 1 представлены данные по вычислению $\|P\|_{B_n}$ для правильного симплекса, вписанного в шар. Мы применяем обозначения раздела 4. Значение k^* равно количеству -1 в экстремальном наборе f_j, соответствующем максимуму в (14). Так как этот максимум оказывается равным $\max\{\psi(a), \psi(a+1)\}$, то k^* совпадает с тем из чисел a или $a+1$, на котором $\psi(t)$ принимает большее значение (см. доказательство теоремы 2). Если $k^*=1$, то в шаре B_n имеется 1-точка относительно

симплекса S, соответствующего проектору P. Для такого n

$$\xi(B_n; S) = \frac{n+1}{2}\Big(\|P\|_{B_n} - 1\Big) + 1, \tag{21}$$

см. (3). Поскольку S — правильный симплекс, вписанный в B_n, то $\xi(B_n; S) = n$ и (21) равносильно $\|P\|_{B_n} = 3 - \frac{4}{n+1}$. Однако, начиная с $n = 5$, всегда $k^* > 1$. Более того, k^* возрастает с ростом n. Это соответствует тому, что (21) и равенство $\|P\|_{B_n} = 3 - \frac{4}{n+1}$ имеют место лишь при $1 \le n \le 4$. Первоначально этот эффект был обнаружен авторами в ходе компьютерных экспериментов и лишь затем было дано аналитическое решение задачи.

Если $n + 1$ есть число Адамара, вопрос о справедливости равенства, аналогичного (21), можно рассмотреть также для n-мерного куба и вписанного в него правильного симплекса. Интересно заметить, что 1-точка Q_n относительно такого S существует не только для $n = 1$ и $n = 3$, но и для $n = 7$. В указанных случаях

$$\xi(Q_n; S) = \frac{n+1}{2}\Big(\|P\|_{Q_n} - 1\Big) + 1,$$

откуда следует, что $\theta_3 = 3$ и $\theta_7 = \frac{5}{2}$. Эта тематика подробно рассмотрена в [4].

Для вычислений и построения графиков в настоящей работе использовалась система Wolfram Mathematica (см. [15, 16]).

Список литературы / References

[1] Невский М. В., “Неравенства для норм интерполяционных проекторов”, *Модел. и анализ информ. систем*, **15**:3 (2008), 28–37; [Nevskij M. V., “Inequalities for the norms of interpolating projections”, *Modeling and Analysis of Information Systems*, **15**:3 (2008), 28–37, (in Russian).]

[2] Невский М. В., “Об одном соотношении для минимальной нормы интерполяционного проектора”, *Модел. и анализ информ. систем*, **16**:1 (2009), 24–43; [Nevskij M. V., “On a certain relation for the minimal norm of an interpolational projection”, *Modeling and Analysis of Information Systems*, **16**:1 (2009), 24–43, (in Russian).]

[3] Невский М. В., “Об одном свойстве n-мерного симплекса”, *Матем. заметки*, **87**:4 (2010), 580–593; English transl.: Nevskii M. V., “On a property of n-dimensional simplices”, *Math. Notes*, **87**:4 (2010), 543–555.

[4] Невский М. В., *Геометрические оценки в полиномиальной интерполяции*, Ярославль: Ярославский государственный университет им. П. Г. Демидова, 2012; [Nevskii M. V., *Geometricheskie ocenki v polinomialnoy interpolyacii*, Yaroslavl: P. G. Demidov Yaroslavl State University, 2012, (in Russian).]

[5] Невский М. В., Ухалов А. Ю., “Новые оценки числовых величин, связанных с симплексом”, *Модел. и анализ информ. систем*, **24**:1 (2017), 94–110; [Nevskii M. V., Ukhalov A. Yu., “New estimates of numerical values related to a simplex”, *Modeling and Analysis of Information Systems*, **24**:1 (2017), 94–110, (in Russian).]; English transl.: Nevskii M. V., Ukhalov A. Yu., “New estimates of numerical values related to a simplex”, *Aut. Control Comp. Sci.*, **51**:7 (2017), 770–782.

[6] Невский М. В., Ухалов А. Ю., “Об оптимальной интерполяции линейными функциями на n-мерном кубе”, *Модел. и анализ информ. систем*, **25**:3 (2018), 291–311; [Nevskii M. V., Ukhalov A. Yu., “On optimal interpolation by linear functions on an n-dimensional cube”, *Modeling and Analysis of Information Systems*, **25**:3 (2018), 291–311, (in Russian).]; English transl.: Nevskii M. V., Ukhalov A. Yu., “On optimal interpolation by linear functions on n-dimensional cube”, *Aut. Control Comp. Sci.*, **52**:7 (2018), 828–842.

[7] Невский М. В., "О некоторых задачах для симплекса и шара в $\mathbb{R}^n$", *Модел. и анализ информ. систем*, **25**:6 (2018), 680–691; [Nevskii M. V., "On some problems for a simplex and a ball in $\mathbb{R}^n$", *Modeling and Analysis of Information Systems*, **25**:6 (2018), 680–691, (in Russian).]

[8] Сегё Г., *Ортогональные многочлены*, Москва: Физматгиз, 1962; [Szegő G., *Orthogonal polynomials*, New York: American Mathematical Society, 1959, (in English).]

[9] Суетин П. К., *Классические ортогональные многочлены*, Москва: Наука, 1979; [Suetin P. K., *Klassicheskie ortogonalnye mnogochleny*, Moskva: Nauka, 1979, in Russian).]

[10] Холл М., *Комбинаторика*, Москва: Мир, 1970; [Hall M., Jr, *Combinatorial theory*, Blaisdall publishing company, Waltham (Massachusets) – Toronto – London, 1967, (in English).]

[11] Hedayat A., Wallis W. D., "Hadamard matrices and their applications", *The Annals of Statistics*, **6**:6 (1978), 1184–1238.

[12] Horadam K. J., *Hadamard matrices and their applications*, Princeton University Press, 2007.

[13] Hudelson M., Klee V., Larman D., "Largest j-simplices in d-cubes: some relatives of the Hadamard maximum determinant problem", *Linear Algebra Appl.*, **241** (1996), 519–598.

[14] Nevskii M., Ukhalov A., "Perfect simplices in $\mathbb{R}^5$", *Beiträge zur Algebra und Geometrie / Contributions to Algebra and Geometry*, **59**:3 (2018), 501–521.

[15] Wellin P., *Essentials of programming in Mathematica*, Cambridge University Press, 2016.

[16] Wolfram S., *An elementary introduction to the Wolfram Language*, Wolfram Media, Inc., 2017.

Nevskii M. V., Ukhalov A. Yu., "Linear Interpolation on a Euclidean Ball in $\mathbb{R}^n$", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 279–296.

DOI: 10.18255/1818-1015-2019-2-279-296

Abstract. For $x^{(0)} \in \mathbb{R}^n, R > 0$, by $B = B(x^{(0)}; R)$ we denote a Euclidean ball in $\mathbb{R}^n$ given by the inequality $\|x - x^{(0)}\| \le R$, $\|x\| := \left(\sum_{i=1}^n x_i^2\right)^{1/2}$. Put $B_n := B(0,1)$. We mean by $C(B)$ the space of continuous functions $f : B \to \mathbb{R}$ with the norm $\|f\|_{C(B)} := \max_{x\in B} |f(x)|$ and by $\Pi_1(\mathbb{R}^n)$ the set of polynomials in n variables of degree ≤ 1, i. e. linear functions on $\mathbb{R}^n$. Let $x^{(1)}, \dots, x^{(n+1)}$ be the vertices of n-dimensional nondegenerate simplex $S \subset B$. The interpolation projector $P : C(B) \to \Pi_1(\mathbb{R}^n)$ corresponding to S is defined by the equalities $Pf\left(x^{(j)}\right) = f\left(x^{(j)}\right)$. Denote by $\|P\|_B$ the norm of P as an operator from $C(B)$ into $C(B)$. Let us define $\theta_n(B)$ as minimal value of $\|P\|_B$ under the condition $x^{(j)} \in B$. In the paper, we obtain the formula to compute $\|P\|_B$ making use of $x^{(0)}$, R, and coefficients of basic Lagrange polynomials of S. In more details we study the case when S is a regular simplex inscribed into B_n. In this situation, we prove that $\|P\|_{B_n} = \max\{\psi(a), \psi(a+1)\}$, where $\psi(t) = \frac{2\sqrt{n}}{n+1}\left(t(n+1-t)\right)^{1/2} + \left|1 - \frac{2t}{n+1}\right|$ $(0 \le t \le n+1)$ and integer a has the form $a = \left\lfloor \frac{n+1}{2} - \frac{\sqrt{n+1}}{2} \right\rfloor$. For this projector, $\sqrt{n} \le \|P\|_{B_n} \le \sqrt{n+1}$. The equality $\|P\|_{B_n} = \sqrt{n+1}$ takes place if and only if $\sqrt{n+1}$ is an integer number. We give the precise values of $\theta_n(B_n)$ for $1 \le n \le 4$. To supplement theoretical results we present computational data. We also discuss some other questions concerning interpolation on a Euclidean ball.

Keywords: n-dimensional simplex, n-dimensional ball, linear interpolation, projector, norm

On the authors:
Mikhail V. Nevskii, orcid.org/0000-0002-6392-7618, Doctor of Science,
P.G. Demidov Yaroslavl State University,
Sovetskaya str., 14, Yaroslavl, 150003, Russian Federation, e-mail: mnevsk55@yandex.ru

Ukhalov Alexey Yurievich, orcid.org/0000-0001-6551-5118, PhD,
P.G. Demidov Yaroslavl State University,
Sovetskaya str., 14, Yaroslavl, 150003, Russian Federation, e-mail: alex-uhalov@yandex.ru

Моделирование и анализ информационных систем. Т. 26, № 2 (2019), с. 297–305
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 297–305

DOI: 10.18255/1818-1015-2019-2-297-305

УДК 517.9

Анализ условий возникновения пространственно-неоднородных структур световых волн в оптических системах передачи информации

Кубышкин Е. П., Куликов В.А.

Поступила в редакцию 24 мая 2019
После доработки 03 июня 2019
Принята к публикации 05 июня 2019

Аннотация. Рассматривается модель распределенных носителей информации в виде устойчивых пространственно-неоднородных структур в системах оптической и волоконно-оптической связи. Изучаются условия возникновения таких устойчивых пространственно-неоднородных структур световой волны генератора оптического излучения. Образование неоднородных структур, которые возникают в плоскости, ортогональной направлению распространения волны, обеспечивается тонким слоем нелинейной среды и контуром двумерной запаздывающей обратной связи с оператором поворота пространственных координат световой волны в плоскости излучения оптического генератора. В пространстве основных параметров генератора (управляющий параметр, угол поворота пространственных координат, величина запаздывания) построены области генерации устойчивых пространственно-неоднородных структур, был проведен анализ механизмов их возникновения.

Ключевые слова: пространственно-неоднородные волны, бифуркация

Для цитирования: Кубышкин Е. П., Куликов В.А., "Анализ условий возникновения пространственно-неоднородных структур световых волн в оптических системах передачи информации", *Моделирование и анализ информационных систем*, **26**:2 (2019), 297–305.

Об авторах:
Кубышкин Евгений Павлович, orcid.org/0000-0003-1796-0190, д-р. физ.-мат. наук, профессор,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: kubysh.e@yandex.ru

Куликов Владимир Александрович, orcid.org/0000-0003-1351-7706, аспирант,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: kulikov7677@gmail.com

Введение

В работе [1] приведены экспериментальные результаты образования пространственно-неоднородных волн в лазерных пучках генератора оптического излучения со специальным контуром обратной связи и предложена математическая модель для описания этого явления, а также результаты ее численного анализа. Математическая модель представляет собой начально-краевую задачу для нелинейного дифференциального уравнения параболического типа с оператором поворота пространственного

аргумента, которое рассматривается в области, определяемой апертурой светового излучения. Эта начально-краевая задача и различные ее обобщения изучались в большом количестве работ (см., например, [2–7]), где различными аналитическими и численными методами строятся пространственно-неоднородные решения. Модель работы [1] не учитывает фактор временного запаздывания в контуре нелинейной обратной связи. Учет временного запаздывания рассмотрен в работе [8], где для математической модели с запаздывающим аргументом в тонкой кольцевой области показана возможность бифуркации из однородного состояния равновесия пространственно-неоднородных бегущих волн. В настоящей работе рассматривается математическая модель работы [1] в круговой области с оператором поворота пространственного аргумента и временным запаздыванием в контуре обратной связи, для которой исследуются условия и характер потери устойчивости однородных состояний равновесия в зависимости от величины запаздывания, возможные бифуркации при потере устойчивости, а также устойчивость пространственно-неоднородных решений. Такие решения могут быть использованы как носители информации в оптической и волоконно-оптической связи.

1. Математическая постановка задачи

В круге $K_R = \{(\rho, \phi) : \ 0 \le \rho \le R, 0 \le \phi \le 2\pi\}$ рассматривается начально-краевая задача вида

$$u_t + u = D\Delta_{\rho\phi}u + K(1 + \gamma \cos u_{\theta T}),\ u_\rho(R, \phi, t) = 0,\ u(\rho, \phi, t) = u(\rho, \phi + 2\pi, t) \quad (1)$$

относительно функции $u(\rho, \phi, t + s), t \ge 0, -T \le s \le 0$, где $T > 0$ величина запаздывания аргумента, с начальным условием $u(\rho, \phi, t+s)|_{t=0} = u_0(\rho, \phi, s) \in H(K_R; -T, 0)$. Пространство начальных условий $H(K_R; -T, 0) = \{u(\rho, \phi, s) : u(\rho, \phi, s) \in C(K_R \otimes [-T, 0])$, при каждом s $u(\rho, \phi, s) \in \overset{\circ}{W}{}_2^1(K_R), u(\rho, \phi, s) = u(\rho, \phi + 2\pi, s)\}$, где пространство функций $\overset{\circ}{W}{}_2^1(K_R)$ получено замыканием множества функций $u(\rho, \phi) \in C^1(K_R), u_\rho(R, \phi) = 0$ в метрике пространства функций $W_2^1(K_R)$. В (1) $\Delta_{\rho\phi}$ – оператор Лапласа в полярных координатах, который считаем симметрично расширенным на пространство $\overset{\circ}{W}{}_2^1(K_R), u_{\theta T}(\rho, \phi, t) \equiv u(\rho, \phi+\theta, t-T)(0 \le \theta < 2\pi)$ – оператор поворота пространственного аргумента и временного запаздывания, D, K – положительные постоянные $0 < \gamma < 1$.

Однородные состояния равновесия $u_* = u_*(K, \gamma)$ начально-краевой задачи (1) определяются как решения уравнения

$$u = K(1 + \gamma cosu). \quad (2)$$

Уравнение (2) в зависимости от K и γ может иметь несколько решений, в том числе кратные. Ниже исследуются условия потери устойчивости состояний равновесия $u_*(K, \gamma)$ и обусловленные ею возможные бифуркации пространственно-неоднородных автоколебательных решений начально-краевой задачи (1).

2. Анализ устойчивости состояний равновесия начально-краевой задачи (1)

Выберем одно из решений $u_*(K,\gamma)$ (не кратное) уравнения (2) и запишем начально-краевую задачу(1) в его окрестности, заменив $u(\rho,\phi,t) \to u_*(K,\gamma)+u(\rho,\phi,t)$. В результате получим начально-краевую задачу

$$u_t+u = D\Delta_{\rho\phi}u - bu_{\theta T}+b_2u_{\theta T}^2+bu_{\theta T}^3/6+\ldots,\ u_\rho(R,\phi,t)=0,\ u(\rho,\phi,t)=u(\rho,\phi+2\pi,t),$$

$$u(\rho,\phi,t+s)|_{t=0}=u_0(\rho,\phi,s), \tag{3}$$

$$b = K\gamma\sin u_*(K,\gamma), \tag{4}$$

где точками обозначены слагаемые, имеющие по $u_{\theta T}$ более высокий порядок малости, $b=K\gamma\sin u_*(K,\gamma)$, $b_2=-K\gamma\cos u_*(K,\gamma)/2$.

Рассмотрим линейную часть начально-краевой задачи (3)

$$u_t+u = D\Delta_{\rho\phi}u - bu_{\theta T},\ u_\rho(R,\phi,t)=0,\ u(\rho,\phi,t)=u(\rho,\phi+2\pi,t),$$

$$u(\rho,\phi,t+s)|_{t=0}=u_0(\rho,\phi,s). \tag{5}$$

Определяя решения (5) вида $u(\rho,\phi,t)=v(\rho,\phi)e^{\lambda t}$, $\lambda\in C$ получим пучок операторов

$$P(\lambda)\equiv\lambda v(\rho,\phi)+v(\rho,\phi)-D\Delta_{\rho,\phi}v(\rho,\phi)+bv(\rho,\phi+\theta)e^{-\lambda T}, \tag{6}$$

действующий в $L_2(K_R)$ с областью определения $\overset{\circ}{W}{}_2^1(K_R)$, спектр которого определяет устойчивость (неустойчивость) решений начально-краевой задачи (5). Определяя $v(\rho,\phi)$ в виде

$$v(\rho,\phi)=v_0+\sum_{j=1}^{\infty}\sum_{n=-\infty}^{\infty}R_{nj}(\rho)(v_{nj}e^{in\phi}+w_{nj}e^{-in\phi}),$$

$$i=\sqrt{-1},\ v_0\in\mathbb{R}, v_{nj},w_{nj}\in\mathbb{C}, v_{-nj}=\bar v_{nj}, w_{-nj}=\bar w_{nj}, v_{0j}=\bar w_{0j},$$

$$R_{nj}(\rho)=\frac{\sqrt{(2)}/RJ_n(\gamma_{nj}\rho/R)}{(1-n^2/\gamma_{jn}^2)^{1/2}J_n^1(\gamma_{nj})},\ (R_{nj}(\rho),R_{np}(\rho))=\int_0^R\rho R_{nj}(\rho)R_{np}(\rho)d\rho=\delta_{jp},$$

где $J_n(\rho)$ функции Бесселя первого рода n-го порядка, γ_{nj} j-й положительный ноль функции $J_n'(\rho)$, а δ_{jp} – символ Кронекера, получим последовательность уравнений

$$(\lambda+1+D\gamma_{nj}^2+be^{in\theta-\lambda T})v_{nj}=0,\quad (\lambda+1+D\gamma_{nj}^2+be^{-in\theta-\lambda T})w_{nj}=0, \tag{7}$$

из которых определяются точки спектра пучка операторов (6), отвечающие за устойчивость решений (5), и ненулевые v_0, v_{nj}, w_{nj}, $n=0,1,2,\ldots$, $j=1,2,\ldots$

Для построения границ областей устойчивости в пространстве параметров D,b,T,θ воспользуемся методом D-разбиений [9,10]. Для этого положим в (7) $\lambda=i\omega$, $\omega\ge 0$ и приравняем нулю вещественную и мнимую части выражений в круглых скобках. В результате получим уравнения

$$\begin{aligned}1+D\gamma_{nj}^2+b\cos(-n\theta+\omega T)=0, \omega-b\sin(-n\theta+\omega T)=0,\\ 1+D\gamma_{nj}^2+b\cos(n\theta+\omega T)=0, \omega-b\sin(n\theta+\omega T)=0.\end{aligned} \tag{8}$$

Эти уравнения позволяют выразить

$$b = b(D,T,n,\omega) = (-1)^{k+1}(1+D\gamma_{nj}^2)/\cos(\mathrm{arctg}(\omega/(1+D\gamma_{nj}^2))),$$

$$\theta^+ = \theta^+(D,T,n,\omega) = (T\omega + \mathrm{arctg}(\omega/(1+D\gamma_{nj}^2)) + \pi k)/n,$$

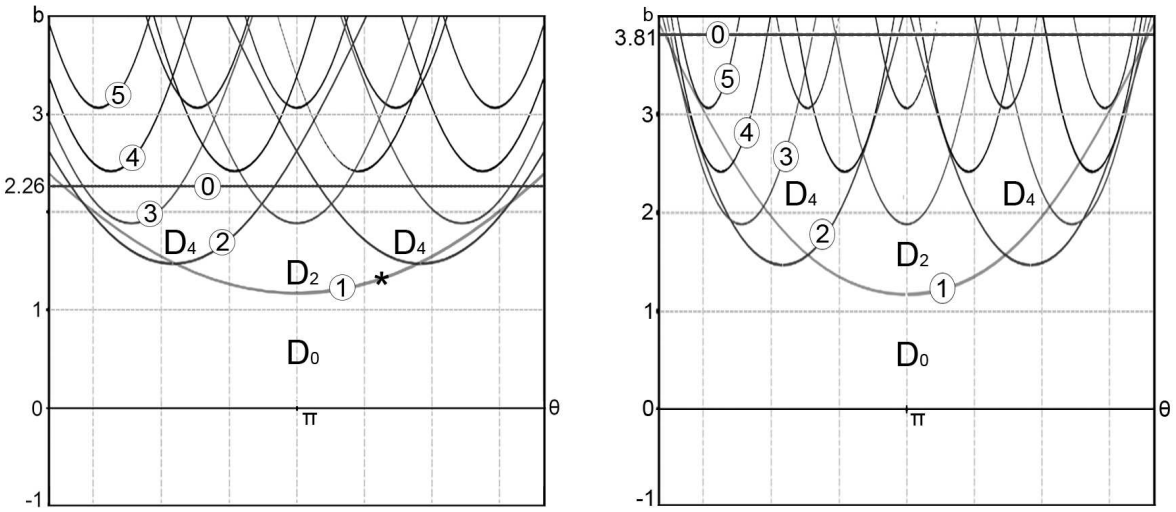
$$\theta^- = \theta^-(D,T,n,\omega) = (-T\omega - \mathrm{arctg}(\omega/(1+D\gamma_{nj}^2)) + \pi k)/n,$$

при $n = 1,2,...,k = 1,2,...,2n-1, j = 1,2,...,$ и

$$b = b(\omega) = (-1)^{k+1}/\cos(\mathrm{arctg}(\omega)),\ T = T(\omega) = \omega^{-1}(\pi k - \mathrm{arctg}(\omega)), k = 1,2,...,$$

при $n = 0$.

Изменяя теперь $0 \le \omega < \infty$ для различных значений D, T, n, построим границы перехода точек спектра пучка операторов (5) из левой комплексной полуплоскости в правую и тем самым получим границу области устойчивости, а также исследуем характер потери устойчивости решений начально-краевой задачи (4).



a) b)

Рис. 1: D-разбиение плоскости параметров (θ, b):
a) D=0.05, T=1.0; b) D=0.05, T=0.5
Fig. 1: D-partition of the parameter plane (θ, b):
a) D=0.05, T=1.0; b) D=0.05, T=0.5

На рисунке 1 для двух значений параметров D и T приведена картина D-разбиения плоскости (θ, b). На рисунках через D_j обозначены области, при значении параметров из которых пучок операторов имеет j точек спектра, принадлежащих правой комплексной полуплоскости, а границы этих областей соответствуют точкам спектра, лежащим на мнимой оси. Из рисунков видно, что имеются значения параметров на границе области устойчивости, при которых в правую полуплоскость может переходить одновременно две пары комплексно сопряженных корней, а также есть на границе области устойчивости значения параметров, соответствующие

двухкратным нулевым точкам спектра пучка операторов (5). Отметим, что граница, соответствующая однородной форме ($n = 0$) потери устойчивости (прямая линия), поднимается вверх при $T \to 0$ и опускается вниз при $T \to \infty$. В связи с этим, при больших значения T невозможна бифуркация пространственно-неоднородных автоколебательных решений.

3. Бифуркация пространственно-неоднородных решений начально-краевой задачи (4)

Выберем параметры Q_*, b_*, D_*, T_* таким образом, чтобы они соответствовали точке границы области устойчивости начально-краевой задачи (4) и при этом пучок операторов (5) имел одну пару комплексно сопряженных точек спектра с нулевой вещественной частью. Пусть для определенности они удовлетворяют первой паре системы уравнений (8). На рисунке 1 это может быть, например, точка, отмеченная *. По b_* выберем k_*, γ_* и $u_* = u(K_*, \gamma_*)$, удовлетворяющие равенствам (2) и (4). Отметим, что такой выбор может быть неоднозначным. Положим теперь $K = K_*(1+\varepsilon)$, где ε – малый параметр, и исследуем возможность бифуркации из состояния равновесия

$$u_*(\varepsilon) = u_*(K_*(1+\varepsilon), \gamma_*) = u_* + \varepsilon u_{*1} + ..., u_{*1} = u_*/(1-b_*)$$

пространственно-неоднородных решений начально-краевой задачи (3) при изменении параметра ε. Для анализа воспользуемся методом инвариантных (центральных) многообразий [11] и теорией нормальных форм обыкновенных дифференциальных уравнений [12].

Отметим, что теперь в (3)

$$b = b(\varepsilon) = K_*(1+\varepsilon)\gamma_* \sin u_*(\varepsilon) = b_* + \varepsilon b_1 + ..., \ b_1 = -b_* + u_*(K_* - u_*)/(1-b_*),$$

$$b_2 = b_2(\varepsilon) = -K_*(1+\varepsilon)\gamma_* \cos u_*(\varepsilon)/2$$

пучок операторов $P(\lambda) \equiv P(\lambda; \varepsilon)$, а также, что $b_* = 1$ возможно лишь для кратного корня u_* уравнения (2). Обозначим через $\lambda(\varepsilon) = i\omega_* + \varepsilon\lambda_1 + ..., \lambda(0) = i\omega_*$ точку спектра оператора $P(\lambda; \varepsilon)$, $\lambda(\varepsilon)$ аналитически зависит от ε.

Начально-краевая задача (4) имеет двумерное пространство решений, которое можно записать в следующей форме:

$$\begin{gathered} u(\rho, \phi, z, \bar{z}; \varepsilon) = R_{n1}(\rho)(e^{in\phi + \lambda(\varepsilon)s} z + e^{-in\phi + \bar{\lambda}(\varepsilon)s}\bar{z}), \\ \dot{z} = \lambda(\varepsilon) z, z = z(t) \in \mathbb{C}. \end{gathered} \tag{9}$$

Остальные решения начально-краевой задачи (4) экспоненциально затухают при $t \to \infty$.

Начально-краевая задача (3) в окрестности нулевого состояния равновесия имеет локальное экспоненциально устойчивое инвариантное многообразие (центральное многообразие), поведение решений на котором определяется поведением решений двумерной системы обыкновенных дифференциальных уравнений.

Для построения центрального многообразия и системы дифференциальных уравнений на нем воспользуемся подходом работы [13] и перейдем от (3) к эквивалентной

начально-краевой задаче в области $(0 \le \rho \le R, 0 \le \phi \le 2\pi, -T_* \le s \le 0), t \ge 0$, положив $w(\rho,\phi,s,t) = u(\rho,\phi,t+s)$:

$$\begin{gathered} w_t = w_s \\ w_s(\rho,\phi,0,t) = -w(\rho,\phi,0,t) + D_*\Delta_{\rho\phi}w(\rho,\phi,0,t) - b(\varepsilon)w(\rho,\phi+\theta_*,-T_*)+ \\ +b_2(\varepsilon)w^2(\rho,\phi+\theta_*,-T_*,t) + b(\varepsilon)w^3(\rho,\phi+\theta_*,-T_*,t)/6 + ..., \end{gathered} \tag{10}$$

$$w_\rho(R,\phi,s,t) = 0, w(\rho,\phi,s,t) = w(\rho,\phi+2\pi,s,t), \tag{11}$$

$$w(\rho,\phi,s,0) = w_0(\rho,\phi,s) \in H(K_R;[-T_*,0]). \tag{12}$$

Будем строить центральное многообразие и дифференциальные уравнения траекторий на нем в виде разложения по

$$\begin{gathered} W(\rho,\phi,s,z,\bar z;\varepsilon) = R_{n1}(\rho)(e^{in\phi+\lambda(\varepsilon)s}z + e^{-in\phi+\bar\lambda(\varepsilon)s}\bar z) + w_{20}(.) + w_{11}(.)z\bar z + \\ +w_{02}(.)\bar z^2 + w_{30}(.)z^3 + w_{21}(.)z^2\bar z + w_{12}(.)\bar z^2 z + w_{03}\bar z^3 + \dots, \\ w_{jk}(.) = \bar w_{jk}(.), w_{jk}(.) = w_{jk}(\rho,\phi,s;\varepsilon), \end{gathered} \tag{13}$$

$$\dot z = \lambda(\varepsilon)z + d_{21}(\varepsilon)z^2\bar z + ... = Z(z,\bar z,\varepsilon). \tag{14}$$

В (13) точками обозначены слагаемые, имеющие более высокий порядок малости по $z, \bar z$. Дифференциальное уравнение для $\bar z$ получается простым сопряжением уравнения для z.

Условие принадлежности траекторий уравнения (14) в силу (13) начально-краевой задаче (10)-(12) дает тождество

$$W_t(.) = W_z(.)Z(z,\bar z;\varepsilon) + W_{\bar z}(.)\bar Z(z,\bar z;\varepsilon) \equiv W_s(.), \tag{15}$$

$$\begin{gathered} W_s(\rho,\phi,0,z,\bar z;\varepsilon) \equiv -W(\rho,\phi,0,z,\bar z;\varepsilon) + D_*\Delta_{\rho\phi}W(\rho,\phi,0,z,\bar z;\varepsilon) - \\ -b(\varepsilon)W(\rho,\phi+\theta_*,-T_*,z,\bar z;\varepsilon) + b_2(\varepsilon)W^2(\rho,\phi+\theta_*,-T_*,z,\bar z;\varepsilon) + \\ +b_3(\varepsilon)W^3(\rho,\phi+\theta_*,-T_*,z,\bar z;\varepsilon)/6 + \dots, \end{gathered} \tag{16}$$

$$W_\rho(R,\phi,s,z,\bar z;\varepsilon) = 0,\ W(\rho,\phi,s,z,\bar z;\varepsilon) = W(\rho,\phi+2\pi,s,z,\bar z;\varepsilon), \tag{17}$$

по переменным $z, \bar z$. При первых степенях оно выполняется в силу (9). Приравняем теперь слева и справа в (15)-(17) коэффициенты при z^2. В результате получим краевую задачу для определения $w_{20}(.)$ вида

$$2\lambda(\varepsilon)w_{20}(.) = w_{20s}(.)$$

$$\begin{gathered} w_{20s}(\rho,\phi,0;\varepsilon) = -w_{20}(\rho,\phi,0;\varepsilon) + D_*\Delta_{\rho\phi}w_{20}(\rho,\phi,0;\varepsilon) - \\ -b(\varepsilon)w_{20}(\rho,\phi+\theta_*,-T_*;\varepsilon) + b_2(\varepsilon)R_{n1}^2(\rho)e^{i(2n\phi+2\theta_*)}e^{-2\lambda(\varepsilon)T_*}, \\ w_{20\rho}(R,\phi,s;\varepsilon) = 0. \end{gathered}$$

Определяя решение $w_{20}(.)$ в виде $w_{20}(\rho,\phi,s,\varepsilon)=v_{20}(\rho,\varepsilon)e^{i2n\phi+2\lambda(\varepsilon)s}$ имеем

$$p_j(\varepsilon)=b_2(\varepsilon)e^{i2-\lambda(\varepsilon)T_*}(R_{n1}^2(\rho),R_{2nj}(\rho))/(2\lambda(\varepsilon)+1+D_*\gamma_{2nj}^2+b(\varepsilon)e^{i2n\theta_*-2\lambda(\varepsilon)T_*}),$$

используя разложение

$$v_{20}(\rho,\varepsilon)=\sum_{j=1}^{\infty}p_j(\varepsilon)R_{2nj}(\rho).$$

Аналогичным образом определяются коэффициенты в разложении при $z\bar{z},z^2$. В результате будем иметь:

$$w_{11}(\rho,\phi,s;\varepsilon)\equiv w_{11}(\rho;\varepsilon)=\sum_{j=0}^{\infty}p_j(\varepsilon)R_{0j}(\varepsilon),$$

$$p_j(\varepsilon)=2b_2(\varepsilon)(R_{n1}^2(\rho),R_{0j}(\rho))/(1+D_*\gamma_{0j}^2+b(\varepsilon)),$$

$$w_{02}(\rho,\phi,s;\varepsilon)=\bar{w}_{20}(\rho,\phi,s;\varepsilon).$$

Рассмотрим определение коэффициента при $z^2\bar{z}$. Из (15)-(17) имеем краевую задачу

$$R_{n1}(\rho)e^{in\phi+\lambda(\varepsilon)s}d_{21}(\varepsilon)+\lambda(\varepsilon)w_{21}(.)=w_{21s}(.),$$

$$w_{21s}(\rho,\phi,0,\varepsilon)=-w_{21}(\rho,\phi,0,\varepsilon)+D_*\Delta_{\rho\phi}w_{21}(\rho,\phi,0;\varepsilon)-$$

$$-b(\varepsilon)w_{21}(\rho,\phi+\theta_*,-T_*;\varepsilon)+[b_2(\varepsilon)(2v_{20}(\rho;\varepsilon)+2v_{11}(\rho;\varepsilon))R_{n1}(\rho)+$$

$$+b(\varepsilon)R_{n1}^3(\rho)/2]e^{in\theta_*-\lambda(\varepsilon)T_*}e^{in\phi},$$

$$w_{21\rho}(R,\phi,s;\varepsilon)=0,w_{21}(\rho,\phi,s;\varepsilon)=w_{21}(\rho,\phi+2\pi,s;\varepsilon).$$

Определяя $w_{21}(\rho,\phi,s;\varepsilon)=(v_{21}(\rho;\varepsilon)+sd_{21}(\varepsilon))e^{in\phi+\lambda(\varepsilon)s},v_{21}(\rho,\varepsilon)=\sum\limits_{j=2}^{\infty}p_j(\varepsilon)R_{nj}(\rho)$, получим, что при j=1 краевая задача неразрешима. Разрешимости добиваемся выбором $d_{21}(\varepsilon)$. В результате имеем:

$$d_{21}(\varepsilon)=e^{in\theta_*-\lambda(\varepsilon)T_*}/(1-T_*b(\varepsilon)e^{in\theta_*-\lambda(\varepsilon)T_*}),$$

$$(2b_2(\varepsilon)(v_{20}(\rho;\varepsilon)+v_{11}(\rho;\varepsilon))R_{n1}(\rho)+b(\varepsilon)R_{n1}^3(\rho)/2,R_{n1}(\rho)).$$

Коэффициенты $p_j(\varepsilon)$ определяются однозначно. Коэффициент $w_{30}(.)$ определяется аналогично квадратичным слагаемым и при этом $w_{12}(.)=\overline{w_{21}}(.),w_{03}(.)=\overline{w_{30}}(.)$. Отметим, что на границе области устойчивости выполняются неравенства $Re\lambda_1>0,Red_{21}(0)<0$. В связи с этим уравнение (14) имеет ассимптотически устойчивое периодическое решение:

$$\begin{gathered}z(t;\varepsilon)=\varepsilon^{1/2}\rho_*e^{i\tau}+O(\varepsilon),\\ \rho_*=(-Re\lambda_1/Red_{21}(0))^{1/2},\\ \dot{\tau}=\omega_*(\varepsilon)=\omega_*+\varepsilon Im\lambda_1+O(\varepsilon^2).\end{gathered}\tag{18}$$

Этому решению в начально-краевой задаче (3) соответствует согласно (13)-(14) ассимптотической устойчивости периодическое решение, которое имеет вид

$$u(\rho, \phi, \tau; \varepsilon) = \varepsilon^{1/2} 2R_{n1}(\rho)\rho_* cos(n\phi + \tau) + O(\varepsilon),$$

где τ определено в (18). Указанное решение аналитически зависит от параметра $\varepsilon^{1/2}$ и является бегущей волной, вращающейся по часовой стрелке с частотой $\omega_*(\varepsilon)$ (спиральной волной).

Список литературы / References

[1] Ахманов С. А., Воронцов М. А., "Неустойчивости и структуры в когерентных нелинейно-оптических системах, охваченных двумерной обратной связью", *Нелинейные волны. Динамика и эволюция. Наука*, 1989, 228–238; [Akhmanov S. A., Vorontsov M. A., "Neustojchivosti i struktury v kogerentnyh nelinejno-opticheskih sistemah, ohvachennyh dvumernoj obratnoj svyaz'yu", *Nauka*, 1989, 228–238, (in Russian).]

[2] Akhmanov S. A., Vorontsov M. A., Ivanov V. Yu., et al., "Controlling transverse wave interactions in nonlinear optics: generation and interaction of spatiotemporal structures", *J. Optical Soc. Amer. Ser. B.*, **9**:1 (1992), 78–90.

[3] Кащенко С. А., "Асимптотика пространственно-неоднородных структур в когерентных нелинейно-оптических системах", *Ж. вычисл. матем. и матем. физ.*, **31**:3 (1991), 467–473; [Kashchenko S. A., "Asimptotika prostranstvenno-neodnorodnyh struktur v kogerentnyh nelinejno-opticheskih sistemah", *ZH. vychisl. matem. i matem. fiz.*, **31**:3 (1991), 467–473, (in Russian).]

[4] Разгулин А. В., "Об автоколебаниях в нелинейной параболической задаче с преобразованным аргументом", *Ж. вычисл. матем. и матем. физ.*, **33**:1 (1993), 69–80; [Razgulin A. V., "Ob avtokolebaniyah v nelinejnoj parabolicheskoj zadache s preobrazovannym argumentom", *ZH. vychisl. matem. i matem. fiz.*, **33**:1 (1993), 69–80, (in Russian).]

[5] Разгулин А. В., "Об одном классе функционально-дифференциальных параболических уравнений нелинейной оптики", *Дифференц. ур-ния.*, **36**:3 (2000), 400–407; [Razgulin A. V., "Ob odnom klasse funkcional'no-differencial'nyh parabolicheskih uravnenij nelinejnoj optiki", *Differenc. ur-niya.*, **36**:3 (2000), 400–407, (in Russian).]

[6] Белан Е. П., "О динамике бегущих волн в параболическом уравнении с преобразованием сдвига пространственной переменной", *Ж. матем. физ., анализа, геометрии.*, **1**:1 (2005), 3–34; [Belan E. P., "O dinamike begushchih voln v parabolicheskom uravnenii s preobrazovaniem sdviga prostranstvennoj peremennoj", *ZH. matem. fiz., analiza, geometrii.*, **1**:1 (2005), 3–34, (in Russian).]

[7] Skubachevskii A. L., "Bifurcation of periodic solutions for nonlinear parabolic functional differential equations arising in optoelectronics", *Nonlinear Analysis: TMA*, **32**:2 (1998), 261–278.

[8] Разгулин А. В., Романенко Т. Е., "Вращающиеся волны в параболическом функционально-дифференциальном уравнении с поворотом пространственного аргумента и запаздыванием,", *Ж. вычисл. матем. и матем. физ.*, **53**:11 (2013), 1804–1821; [Razgulin A. V., Romanenko T. E., "Vrashchayushchiesya volny v parabolicheskom funkcional'no-differencial'nom uravnenii s povorotom prostranstvennogo argumenta i zapazdyvaniem", *ZH. vychisl. matem. i matem. fiz.*, **53**:11 (2013), 1804–1821, (in Russian).]

[9] Неймарк Ю. И., "D-разбиение пространства квазиполиномов (к устойчивости линеаризованных распределенных систем)", *ПММ*, **13**:4 (1949), 349–380; [Neimark Yu. I., "Drazbienie prostranstva kvazipolinomov (k ustoychivosti linearizovannykh raspredelennykh sistem)", *PMM*, **13**:4 (1949), 349–380, (in Russian).]

[10] Neimark Yu. I., "The structure of the D-decomposition of the space of quasipolynomials and the diagrams of Vyšnegradskii and Nyquist", *Dokl. Akad. Nauk USSR*, **60** (1948), 1503–1506.

[11] Марсден Дж., Мак-Кракен М., "Бифуркация рождения цикла и ее приложения", *Мир*; [Marsden J. E., McCracken M., "The Hopf bifurcation and its applications", 1948]

[12] Брюно А. Д., "Локальный метод нелинейного анализа дифференциальных уравнений", *Наука*, 1979; [Bryuno A. D., "Lokal'nyj metod nelinejnogo analiza differencial'nyh uravnenij", *Naika*, 1979, (in Russian).]

[13] Kubyshkin E. P., Moriakova A. R., "Features of Bifurcations of Periodic Solutions of the Ikeda Equation. Russian Journal of Nonlinear Dynamics", *Dokl. Akad. Nauk USSR*, **14**:3 (2018), 301–324.

Kubyshkin E. P., Kulikov V.A., "Analysis of the Conditions for the Emergence of Spatially Inhomogeneous Structures of Light Waves in Optical Information Transmission Systems", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 297–305.

DOI: 10.18255/1818-1015-2019-2-297-305

Abstract. A model of distributed information carriers in the form of stable spatially inhomogeneous structures in optical and fiber-optic communication systems is considered. We study the conditions for the occurrence of such stable spatially inhomogeneous structures of the light wave of the generator of optical radiation. The formation of inhomogeneous structures that occur in a plane orthogonal to the direction of wave propagation is provided by a thin layer of nonlinear medium and a two-dimensional lagging feedback loop with the rotation operator of the spatial coordinates of the light wave in the emission plane of the optical generator. In the space of the main parameters of the generator (a control parameter, the angle of rotation of the spatial coordinates, the magnitude of the delay), the areas of generation of stable spatially inhomogeneous structures are constructed, the mechanisms of their occurrence are analyzed.

Keywords: spatially inhomogeneous waves, bifurcation

On the authors:
Evgenii P. Kubyshkin, orcid.org/0000-0003-1796-0190, Doctor, Professor,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: kubysh.e@yandex.ru

Vladimir A. Kulikov, orcid.org/0000-0003-1351-7706, graduate student,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: kulikov7677@gmail.com

Моделирование и анализ информационных систем. Т. 26, №2 (2019), с. 306–311
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 306–311

Algorithms

DOI: 10.18255/1818-1015-2019-2-306-311

УДК 510.5

eT-сводимость множеств

Яруллин Р. Р.

Поступила в редакцию 20 декабря 2018
После доработки 15 мая 2019
Принята к публикации 17 мая 2019

Аннотация. Статья посвящена eT-сводимости – самой общей в интуитивном смысле алгоритмической сводимости, являющейся одновременно сводимостью по перечислимости и сводимостью по разрешимости. Рассматривается соответственная степенная структура – верхняя полурешётка eT-степеней. Показано, что на ней можно корректным образом определить операцию скачка, используя T-скачок или e-скачок множеств. Рассмотрены локальные свойства eT-степеней: тотальность и перечислимость. Доказано, что все степени между наименьшим элементом и первым скачком в $\mathbf{D_{eT}}$ являются вычислимо перечислимыми, более того, эти степени содержат вычислимо перечислимые множества и только их. Установлено существование нетотальных eT-степеней. На основе этого получены некоторые результаты о соотношениях между степенями, в частности, из того, что всякая eT-степень или содержится полностью в некоторой T- или e-степени, или полностью совпадает с ней, следует, что нетотальные e-степени, содержащиеся в T-степенях, расположенных выше второго T-скачка, совпадают с соответствующими нетотальными eT-степенями.

Ключевые слова: eT-сводимость, eT-степени, eT-скачок

Для цитирования: Яруллин Р. Р., "eT-сводимость множеств", *Моделирование и анализ информационных систем*, **26**:2 (2019), 306–311.

Об авторах:
Яруллин Роман Ревович, orcid.org/0000-0003-1604-5599, аспирант,
Ивановский государственный университет,
ул.Ермака, 99, г. Иваново, 153025 Россия, e-mail: iarul1402@yandex.ru

1. Определение eT-сводимости и eT-степеней

Будем придерживаться терминологии и обозначений, используемых в монографиях [1] и [2].

Пусть $\mathbb{N} = \{0, 1, 2, \ldots\}$ – множество натуральных чисел; A, B, $C \subseteq \mathbb{N}$; $\overline{A} = \mathbb{N} - A$; u, x, y, $z \in \mathbb{N}$; φ – частичная функция $\varphi : \mathbb{N} \to \mathbb{N}$, $\operatorname{dom}\varphi$ – её область определения, $\operatorname{ran}\varphi$ – область значений, $\operatorname{graph}\varphi = \{\langle x, y\rangle : \varphi(x) = y\}$ – её график, где $\langle x, y\rangle$ – канторов номер упорядоченной пары (x, y); f – тотальная функция, т. е. $\operatorname{dom}\varphi = \mathbb{N}$; φ_z – частично вычислимая (всюду далее – ч. в.) функция с гёделевым номером z; φ_z^A – функция, ч. в. с использованием оракула для множества A, имеющая гёделев номер z; W_z – вычислимо перечислимое (всюду далее – в. п.) множество с индексом z;

D_u – конечное множество с каноничным индексом u; c_A и χ_A – характеристическая функция и частичная характеристическая функция множества A.

Под сводимостью будем понимать всякое рефлексивное и транзитивное бинарное отношение на $2^{\mathbb{N}}$. Пусть r – некоторая сводимость. Тот факт, что A r-сводится к B будет обозначаться через $A \leqslant_r B$. Приведем ряд связанных определений:

$A \equiv_r B \Longleftarrow A \leqslant_r B \wedge B \leqslant_r A$;

$A <_r B \Longleftrightarrow A \leqslant_r B \wedge B \not\leqslant_r A$;

$\deg_r A := \{X : X \equiv_r A\}$ – r-степень множества A;

$\mathbf{a_r}$, $\mathbf{b_r}$ – произвольные r-степени;

$\mathbf{D_r}$ – класс всех r-степеней.

На $\mathbf{D_r}$ естественным образом вводится частичный порядок. Пусть $A \in \mathbf{a_r}$, $B \in \mathbf{b_r}$, тогда

$$\mathbf{a_r} \leqslant \mathbf{b_r} \Longleftrightarrow \deg_r A \leqslant \deg_r B \Longleftrightarrow A \leqslant_r B.$$

Наибольший интерес представляют сводимости, у которых есть понятная интуитивная подоплека. Среди таковых выделяют T-сводимость и e-сводимость.

Интуитивно, A e-сводится к B, если существует алгоритм, который по любому перечислению множества B даёт некоторое перечисление A, и A T-сводится к B, если существует алгоритм, сводящий проблему разрешения множества A к проблеме разрешения для множества B.

Приведем формальные определения в том виде, в каком они даны в [1].

$$A \leqslant_e B \Longleftrightarrow \exists z \forall x [x \in A \Rightarrow \exists u [\langle x, u\rangle \in W_z \wedge D_u \subseteq B]],$$
$$A \leqslant_T B \Longleftrightarrow \exists z [c_A = \varphi_z^B].$$

Если фиксировать z в определении e-сводимости, то можно определить некоторый оператор Φ_z, такой что

$$\Phi_z(X) = \{x : \exists u [\langle x, u\rangle \in W_z \wedge D_u \subseteq X]\} \quad \text{и} \quad A = \Phi_z(B),$$

который будем называть оператором перечисления.

Про две сводимости $\leqslant_{r_1}$ и $\leqslant_{r_2}$ говорят, что $\leqslant_{r_1}$ сильнее $\leqslant_{r_2}$, а $\leqslant_{r_2}$ слабее $\leqslant_{r_1}$, если имеет место следующая импликация

$$\forall A, B \quad [A \leqslant_{r_1} B \Rightarrow A \leqslant_{r_2} B].$$

Если некоторая сводимость сильнее T-сводимости, то она называется сводимостью по разрешимости, а если она сильнее e-сводимости, то сводимостью по перечислимости. T-сводимость и e-сводимость несравнимы друг с другом в отношении сильнее/слабее. Тем самым, T-сводимость – самая слабая из сводимостей по разрешимости, а e-сводимость – самая слабая из сводимостей по перечислимости, а обе они – самые «широкие» в интуитивном смысле сводимости.

Далее будем рассматривать конъюнкцию этих двух сводимостей – eT-сводимость. Впервые эта сводимость была рассмотрена Ходжаянцем М. Ю. в [3]. Очевидно, что

эта сводимость является сводимостью и по разрешимости, и по перечислимости, более того, эта самая слабая из сводимостей, являющихся сводимостями и по разрешимости, и по перечислимости одновременно.

Хорошо изучены 1-, m- и pm-сводимости также являющиеся одновременно сводимостями по перечислимости и по разрешимости (см. [4]). Легко убедиться в следующем

$$\forall A, B \quad [A \leqslant_1 B \Rightarrow A \leqslant_m B \Rightarrow A \leqslant_{pm} B \Rightarrow A \leqslant_{eT} B].$$

Определение 1. $A \leqslant_{eT} B \iff A \leqslant_e B \wedge A \leqslant_T B$.

Очевидно, что отношение $\leqslant_{eT}$ рефлексивно и транзитивно, и в этом смысле приведённое определение eT-сводимости является корректным.

Теорема 1. $\deg_{eT} A = \deg_e A \cap \deg_T A$.

Доказательство.

$$\begin{aligned}\deg_{eT} A &= \{X : X \equiv_{eT} A\} = \{X : X \equiv_e A \wedge X \equiv_T A\} = \\ &= \{X : X \equiv_e A\} \cap \{X : X \equiv_T A\} = \deg_e A \cap \deg_T A.\end{aligned}$$ □

Таким образом, всякая eT-степень – это пересечение некоторых e- и T–степеней. Как известно, существуют несравнимые множества относительно e- и T-сводимости (см. например [5]), откуда сразу по определению 1 получается следующее предложение.

Предложение 2. $\exists A, B \quad [A \not\leqslant_{eT} B \wedge B \not\leqslant_{eT} A]$.

2. О структуре класса eT-степени. В. п. и тотальные eT-степени

Предложение 3. $\mathbf{D_{eT}}$ – верхняя полурешётка с наименьшим элементом $\mathbf{0_{eT}}$, состоящим из всех вычислимых множеств.

Доказательство. Как известно, $A \oplus B = \{2x : x \in A\} \cup \{2x+1 : x \in B\}$ – наименьшая верхняя грань для любых A и B как по e-сводимости, так и по T-сводимости (см. например [4, с. 38; с. 67]. Из этого следует, что $A \oplus B$ будет также наименьшей верхней гранью для любых A и B по eT-сводимости, т. е. $\mathbf{D_{eT}}$ есть верхняя полурешётка.

Пусть $\mathbf{0_e}$ и $\mathbf{0_T}$ – наименьшие элементы в $\mathbf{D_e}$ и $\mathbf{D_T}$ соответственно. Как известно, $\mathbf{0_e}$ состоит точно из всех в. п. множеств, а $\mathbf{0_T}$ точно из всех вычислимых множеств, т. е. $\mathbf{0_T} \subset \mathbf{0_e}$. Отсюда, в силу теоремы 1, следует, что $\mathbf{0_{eT}} = \mathbf{0_T} \cap \mathbf{0_e} = \mathbf{0_T}$. Таким образом, $\mathbf{0_{eT}}$ состоит из всех вычислимых множеств. □

Определение 2. r-степень называется в. п., если она содержит в. п. множество.

Основные результаты по изучению в. п. T-степеней изложены в [2]. В $\mathbf{D_e}$ существует всего одна в. п. степень – это $\mathbf{0_e}$. Довольно интересные результаты о в. п. степенях в $\mathbf{D_{eT}}$ даёт следующая теорема.

Теорема 4.

(i) Если eT-степень содержит в. п. множество, то она вся состоит из в. п. множеств.

(ii) Существует счётная возрастающая цепь в. п. eT-степеней.

(iii) Существует счётная антицепь в. п. eT-степеней.

Доказательство. (i) $\deg_{eT} W_z = \deg_e W_z \cap \deg_T W_z$, а $\deg_e W_z$ состоит точно из всех в. п. множеств.

(ii), (iii) Следует из аналогичных результатов для T-степеней (см. [1, с. 216]), достаточно лишь взять eT-степени соответствующих в. п. множеств. □

Для получения дальнейших результатов о $\mathbf{D_{eT}}$ введем определение скачка на степенях.

Определение 3. r-скачком множества A называется унарная операция J_r такая, что

(a) $A <_r J_r(A)$;

(b) $A \equiv_r B \Rightarrow J_r(A) \equiv_r J_r(B)$.

Из определения r-скачка для множеств видно, что для каждой $\mathbf{a_r}$ r-степень множества $J_r(A)$ не зависит от выбора множества $A \in \mathbf{a_r}$, что естественным образом позволяет перенести определение скачка для множеств на r-степени. Скачком $\mathbf{a_r}$ будем называть r-степень $\mathbf{j}(\mathbf{a_r}) = \deg_r J_r(A)$.

В качестве определений для T-скачка и e-скачка множеств будем использовать следующие:

$$J_T(A) := K_A = \{x : \varphi_x^A \downarrow\}, \qquad J_e(A) := (A^e \oplus \overline{A^e})^e.$$

Здесь $A^e = \{\langle x, y\rangle : x \in \Phi_y(A)\}$ – e-цилиндр множества A. Данное определение e-скачка было дано Розинасом в [7].

Напомним, что $A \leqslant_1 B$, если $\exists f$ – вычислимая биективная функция такая, что $x \in A \Leftrightarrow f(x) \in B$.

Приведем несколько важных нам свойств, определённых выше e-скачка и T-скачка множеств. $\forall A, B$

(i) $A <_1 J_e(A)$, см. в [4, с. 55];

(ii) $A \equiv_e B \Leftrightarrow J_e(A) \equiv_1 J_e(B)$, см. в [4, с. 55];

(iii) $A \leqslant_T B \Rightarrow A \leqslant_1 J_T(B), \overline{A} \leqslant_1 J_T(B)$, см. в [1, с. 327–328];

(iv) $A \leqslant_1 J_T(A), \overline{A} \leqslant_1 J_T(\overline{A})$;

(v) $A \equiv_T B \Leftrightarrow J_T(A) \equiv_1 J_T(B)$, см. в [1, с. 327–328].

Из перечисленных свойств в силу импликации $A \leqslant_1 B \Rightarrow A \leqslant_{eT} B$ получается следующая теорема.

Теорема 5. e-скачок и T-скачок множеств являются корректными определениями для eT-скачка множеств. Иными словами:

$$\forall A, B \quad A <_{eT} J_T(A), \quad A <_{eT} J_e(A),$$
$$A \equiv_{eT} B \Rightarrow J_T(A) \equiv_{eT} J_T(B), \quad A \equiv_{eT} B \Rightarrow J_e(A) \equiv_{eT} J_e(B).$$

Теорема 6. Пусть $\mathbf{j_T}(\mathbf{a_{eT}})$ – T-скачок eT-степени $\mathbf{a_{eT}}$. Всякая eT-степень $\mathbf{a_{eT}}$ такая, что $\mathbf{0_{eT}} \leqslant \mathbf{a_{eT}} \leqslant \mathbf{j_T}(\mathbf{a_{eT}})$ является в. п. степенью. Таким образом, в силу теоремы 4 все множества, расположенные между $\mathbf{0_{eT}}$ и $\mathbf{j_T}(\mathbf{0_{eT}})$, в. п.

Доказательство. Предположим, что в указанном промежутке существует степень $\mathbf{a_{eT}}$, которая не является в. п, и пусть $A_1 \in \mathbf{a_{eT}}$. Пусть $A \in \mathbf{j_T}(\mathbf{0_{eT}})$, тогда $A_1 \leqslant_{eT} A$, откуда $A_1 \leqslant_e A$, но A – в. п. множество в силу теоремы 4 и того факта, что $K \in \mathbf{j_T}(\mathbf{0_{eT}})$, что следует из определения T-скачка. Таким образом, получаем противоречие, т. к. степень $\mathbf{0_e}$ наименьшая и состоит из всех в. п. множеств. □

Определение 4. r-степень называется тотальной, если она содержит график тотальной функции.

Для T-степеней вопрос об их тотальности решается тривиально: всякая T-степень $\mathbf{a_T}$ тотальна, т. к. вместе с некоторым множеством A содержит график функции c_A. Для e-степеней всё не так однозначно: Медведевым Ю. Т. в [8] было построено такое множество A, что для всякой тотальной функции f, если $\operatorname{graph} f \leqslant_e A$, то f – вычислимое множество. Степень такого множества A называется квазиминимальной. Очевидно, что e-степень такого множества нетотальна. Таким образом, переходим к следующей теореме.

Теорема 7. Существуют нетотальные eT-степени.

В [9] было показано, что внутри каждой T-степени $\geqslant \mathbf{j_T}(\mathbf{j_T}(\mathbf{0_T}))$ содержится некоторая e-степень, а для такой e-степени существует подходящая eT-степень такая, что имеет место $\mathbf{a_e} = \mathbf{a_{eT}}$. А если использовать результат, полученный в [10], то имеем, что все нетотальные e-степени, содержащиеся в T-степенях $\geqslant \mathbf{j}(\mathbf{j}(\mathbf{0_T}))$, полностью совпадают с соответствующими нетотальными eT-степенями.

В [10] был получен результат, что никакая тотальная e-степень не содержится ни в какой T-степени. В силу теоремы 1 для eT-степеней это не так, т. к. всякая eT-степень содержится внутри некоторой T-степени.

Список литературы / References

[1] Роджерс Х., *Теория рекурсивных функций и эффективная вычислимость*, М.: Мир, 1972; [Rogers H., *Theory of Recursive Functions and Effective Computability*, The MIT Press, 1987, (in Russian).]

[2] Соар Р. И., *Вычислимо перечислимые множества и степени*, Казань: Казанское математическое общество, 2000; [Soare Robert I., *Recursively Enumerable Sets and Degrees*, Springer, 1999].

[3] Ходжаянц М. Ю., "О структуре e-степеней", *Известия АН АрССР «Математика»*, **XV**:№ 3 (1980), 165–175; [Hodzhayanc M. YU., "O strukture e-stepenej", *Izvestiya AN ArSSR "Matematika"*, **XV**:3 (1980), 165–175].

[4] Поляков Е. А., Розинас М. Г., *Теория алгоритмов*, Иваново: Из-во ИВГУ, 1976; [Polyakov E. A., Rozinas M. G., *Teoriya algoritmov*, Ivanovo: IVGU, 1976].

[5] Kleene S. C., Post E. L., "The upper semi-lattice of degrees of recursive unsolvability", *Annals of Mathematics*, **59** (1954), 379–407.

[6] Case J., "Enumeration reducibility and partial degrees", *Annals of Mathematical Logic*, **2**:4 (1971), 419–439.

[7] Розинас М. Г., "Операция скачка для некоторых видов сводимости", *ВИНИТИ Деп. 3185-76*; [Rozinas M. G., "Operaciya skachka dlya nekotoryh vidov svodimosti", *VINITI Dep. 3185-76].*

[8] Медведев Ю. Т., "Степени трудности массовых проблем", *Докл. АН СССР*, **104** (1955), 501–504; [Medvedev YU. T., "Stepeni trudnosti massovyh problem", *Dokl. AN SSSR*, **104** (1955), 501–504].

[9] Ходжаянц М. Ю., "e-степени, T-степени и аксиоматические теории", *ДАН АрмССР*, **73**:2 (1981), 73–77; [Hodzhayanc M. Yu., "e-stepeni, T-stepeni i aksiomaticheskie teorii", *DAN ArSSR*, **73**:2 (1981), 73–77].

[10] Солон Б. Я., "Соотношения между e-степенями и T-степенями", *Известия высших учебных заведений, "Математика"*, **3** (1995), 51–61; [Solon B. YA., "O sootnoshenie mezhdu e-stepenyami i T-stepenyami", *Izvestiya vuzov, "Matematika"*, **3** (1995), 51–61].

Iarullin R. R., "eT-reducibility of Sets", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 306–311.

DOI: 10.18255/1818-1015-2019-2-306-311

Abstract. This paper is dedicated to the study of eT-reducibility — the most common in the intuitive sense of algorithmic reducibility, which is both enumeration reducibility and decidable one. The corresponding structure of degrees — upper semilattice of eT-degrees is considered. It is shown that it is possible to correctly define the jump operation on it by using the T-jump or e-jump of sets. The local properties of eT-degrees are considered: totality and computably enumerable. It is proved that all degrees between the smallest element and the first jump in $\mathbf{D_{eT}}$ are computably enumerable, moreover, these degrees contain computably enumerable sets and only them. The existence of non-total eT-degrees is established. On the basis of it, some results have been obtained on the relations between, in particular, from the fact that every eT-degree is either completely contained in some T- or e-degrees, or completely coincides with it, it follows that non-total e-degrees contained in the T-degrees, located above the second T-jump, coincide with the corresponding non-total eT-degrees.

Keywords: eT-reducibility, eT-degrees, eT-jump

On the authors:
Roman R. Iarullin, orcid.org/0000-0003-1604-5599, graduate student,
Ivanovo State University,
99 Ermaka str., Ivanovo, 153025 Russia, e-mail: iarul1402@yandex.ru