

ISSN 2222-8772

**МАТЕМАТИЧЕСКИЕ
СТРУКТУРЫ
И
МОДЕЛИРОВАНИЕ**

**№1 (77)
2026**



**МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Ф.М. ДОСТОЕВСКОГО»**

**МАТЕМАТИЧЕСКИЕ
СТРУКТУРЫ
И
МОДЕЛИРОВАНИЕ**

№ 1 (77)

**Омск
2026**

Редакционная коллегия

А. К. Гуц	главный редактор , председатель редакционной коллегии, доктор физ.-мат. наук, профессор, заслуженный профессор Омского государственного университета им. Ф. М. Достоевского, профессор кафедры информационных технологий и математики Сочинского государственного университета, г. Сочи
Н. Ф. Богаченко	зам. гл. редактора , технический редактор, канд. физ.-мат. наук, доцент, зав. кафедрой компьютерных технологий и сетей Омского государственного университета им. Ф. М. Достоевского, г. Омск
Д. Н. Лавров	зам. гл. редактора , выпускающий редактор, канд. техн. наук, доцент, проректор по образовательной деятельности и информатизации Нижневартковского государственного университета, г. Нижневартовск
В. В. Варламов	доктор физ.-мат. наук, профессор кафедры прикладной математики и информатики Сибирского государственного индустриального университета, г. Новокузнецк
В. П. Голубятников	доктор физ.-мат. наук, профессор, главный научный сотрудник Института математики СО РАН им. С. Л. Соболева, г. Новосибирск
С. И. Горлов	доктор физ.-мат. наук, профессор, ректор Нижневартковского государственного университета, г. Нижневартовск
А. Г. Гринь	доктор физ.-мат. наук, профессор, профессор кафедры информационной безопасности Омского государственного университета им. Ф. М. Достоевского, г. Омск
В. А. Еровенко	доктор физ.-мат. наук, профессор, г. Минск, Республика Беларусь
V. Zilber	Dr.Sc. (Phys.-Math.), Professor of Mathematical Logic, Mathematical Institute, University of Oxford, United Kingdom
А. Н. Кабанов	канд. физ.-мат. наук, доцент, декан факультета информационных технологий и математики Нижневартковского государственного университета, г. Нижневартовск
А. В. Копыльцов	доктор техн. наук, профессор, профессор кафедры информационных систем Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В. И. Ульянова (Ленина), г. Санкт-Петербург
А. Г. Коробейников	доктор техн. наук, профессор, зам. директора по науке Санкт-Петербургского филиала Института земного магнетизма, ионосферы и распространения радиоволн им. Н. В. Пушкина РАН, г. Санкт-Петербург
П. А. Корчагин	доктор техн. наук, профессор, ректор Омского государственного технического университета, г. Омск
V. Kreinovich	Ph.D. (Phys.-Math.), Professor, Computer Science Department, University of Texas at El Paso, Texas, USA
В. А. Плетюхов	доктор физ.-мат. наук, профессор, профессор кафедры общей и теоретической физики Брестского государственного университета им. А. С. Пушкина, г. Брест, Республика Беларусь
Л. Б. Соколинский	доктор физ.-мат. наук, профессор, зав. кафедрой системного программирования Южно-Уральского государственного университета (национального исследовательского университета), г. Челябинск
A. A. Fedorenko	Ph.D. (Phys.-Math.), Researcher (CR1) at the French National Centre of Scientific Research (CNRS) Laboratoire de Physique de l'ENS-Lyon, France
A. Jadczyk	Ph.D., Professor, Researcher, Laboratoire de Physique, Universite de Toulouse III et CNRS, France, and Ronin Institute, Montclair, New Jersey, USA

Учредитель

Федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный университет им. Ф. М. Достоевского»

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций.
Реестровая запись ПИ № ФС77-72200 от 15 января 2018 г.

Адрес редакции, издателя и типографии

644077, Омская обл., г. Омск,
пр-т Мира, д. 55а.

Дата выхода в свет: 7.05.2026.

Тираж 25 экз.

Свободная цена.

МАТЕМАТИЧЕСКИЕ СТРУКТУРЫ И МОДЕЛИРОВАНИЕ

Журнал основан в 1998 году. В журнале публикуются статьи, в которых излагаются результаты исследований по математике, теоретической физике, компьютерным наукам, философии и истории математики и информатики.

Все статьи журнала проходят обязательное рецензирование. Рефераты статей журнала опубликованы в «Реферативном журнале», «Mathematical Reviews» (США), «Zentralblatt für Mathematik» (Германия) и в РИНЦ (elibrary.ru). Журнал входит в Перечень рецензируемых научных изданий ВАК РФ, входящих в Международные реферативные базы данных (МРБД) и системы цитирования (Список от 31.12.2023, № 769), в которых должны быть опубликованы основные результаты диссертаций на соискание учёных степеней (Приказ Минобрнауки России от 25.07.2014 № 793). Имеет категорию K1.

Все статьи в журнале публикуются под лицензией Attribution 4.0 International (CC-BY).

Электронная версия журнала представлена в сети:

<http://msm.omsu.ru>

<http://msm.univer.omsk.su>

Электронная почта выпускающего редактора:

dmtrl@ya.ru

СОДЕРЖАНИЕ

Александр Константинович Гуц (30.08.1947 – 05.03.2026) 5

Фундаментальная математика и физика

Р.Н. Баротов, Д.Н. Баротов. *Об одном гибридном способе решения сложной системы из десяти уравнений* 8

В.В. Варламов. *Теоретико-групповое описание не-периодической системы элементов: VIII. Структура уровней энергии* 17

А.К. Гуц. *Логическое доказательство разрешимости парадокса дедушки в рамках модальной предикатной логики Гёделя S5 второго порядка* . . . 34

Д.А. Друзьев, О.А. Терентьева, П.В. Прудников. *Генерация случайных состояний в спиновых системах с непрерывной симметрией: аспекты применения алгоритма Ванга–Ландау* 37

В.И. Субботин. *К доказательству леммы о локальном вращении* 44

Н.Л. Шаламова. *Мировые линии световых частиц на «конусах прошлого» для некоторых нерелятивистских порядков в R^3* 48

Прикладная математика и моделирование

Ю.В. Захарова, В.П. Лобанов, Р.Г. Кравцов. *Эволюционные алгоритмы для оптимизационной версии задачи граф-подграф изоморфизма* 53

Информационная безопасность

Д.М. Арушанян, С.В. Усов. *О возможности управления ролевой политикой безопасности полномочиями Take и Grant* 63

Продолжение на следующей странице

Наши публикации



Компьютерные науки

- Т.Е. Болдовская, М.И. Ремизов, А.Е. Ветров.
Управление слоями геопространственных данных через справочник типов сущностей 69
- И.Ф. Горкун. *Оконная модель и рекуррентный алгоритм агрегирования транзакций в распределённой биллинговой подсистеме* 78
- Т.В. Костеннов. *Использование синтетических данных и ко-симуляции AnyLogic–Python для разработки GRU-MPC алгоритмов управления микроклиматом теплиц* 88
- О.А. Терентьева. *Автоматизированное прогнозирование постпожарной сукцессии лесов на основе марковских моделей* 102
- В.В. Фирстов. *Система профилирования Unity3D с формированием датасетов для ML-оптимизации* 110
- С.Н. Чуканов, И.С. Чуканов, С.В. Лейхтер. *Сравнение временных рядов на основе анализа соответствующих графов* 121

Памяти Александра Константиновича Гуца

30.08.1947 – 05.03.2026



5 марта 2026 года на 79-м году жизни после тяжелой болезни скончался Александр Константинович Гуц – доктор физико-математических наук, профессор, заслуженный работник высшей школы Российской Федерации, основатель факультета компьютерных наук Омского государственного университета им. Ф.М. Достоевского и бессменный главный редактор журнала «Математические структуры и моделирование».

Ученик академической школы

Александр Константинович родился 30 августа 1947 года в Берлине в семье военнослужащего. В 1970 году окончил механико-математический факультет Новосибирского государственного университета, а в 1973-м – аспирантуру НГУ под руководством выдающегося математика, академика АН СССР Александра Даниловича Александрова. Став представителем Ленинградской геометрической школы, он защитил кандидатскую диссертацию по геометрии и топологии (1973), а в 1987 году – докторскую по той же специальности.

Научное наследие

Научные интересы Александра Константиновича охватывали поразительно широкий спектр – от фундаментальных проблем физики до прикладных задач стоматологии.

Он решил две проблемы, поставленные его учителем А.Д. Александровым. Первая относилась к хроногеометрии – аксиоматической теории относительности. Гуц одним из первых описал хроногеометрию пространств с некоммутативной группой преобразований и представил аксиоматическую теорию стационарной вселенной де Ситтера. Вторая проблема касалась описания физических условий, при которых возможны временные петли – знаменитая машина времени Курта Гёделя. Учёный полностью описал условия существования природной машины времени в пространстве-времени с евклидовой топологией, показал, какие энергии требуются для изменения размерности пространства-времени и порождения кротовых нор. Он построил теорию квантового механизма межвременных переходов, создав негёделевскую модель машины времени.

Эти фундаментальные исследования нашли отражение в монографиях, ставших классическими: «Элементы теории времени», «Хроногеометрия. Аксиоматическая теория относительности», «Физика реальности», «Время. Машина времени. Параллельные вселенные».

С 1989 года, возглавив кафедру математического моделирования, Александр Константинович обратился к практическим задачам. Он создал математическую теорию этногенеза, разработал методы прогнозирования этносоциальных отношений. В сотрудничестве с профессором стоматологии В.М. Семенюком они первыми в 1990-е годы применили компьютерное моделирование в ортопедической стоматологии – по их методикам обучали студентов Омской медицинской академии.

В 2010-х годах Гуц использовал теорию катастроф и теорию игр для моделирования экологических проблем: подтопления лесов, пожаров, управления лесными хозяйствами.

Создание факультета и подготовка кадров

В 2001 году Александр Константинович основал факультет компьютерных наук ОмГУ (ныне – факультет цифровых технологий, математики и кибербезопасности) и руководил им до 2022 года.

Его организаторский талант раскрылся в открытии новых направлений подготовки, определивших развитие ИТ-образования в регионе:

- специальность «Компьютерная безопасность»;
- специальность «Вычислительные машины, комплексы, системы и сети»;
- направление «Информатика и вычислительная техника»;
- направление «Информационная безопасность»;
- направление «Прикладная информатика».

Он также организовал подготовку социологов на факультете компьютерных наук, читая курсы по математическим методам в социологии, математической логике и моделированию социальных систем.

В 1998 году Александр Константинович основал научный журнал «Математические структуры и моделирование» и оставался его бессменным главным редактором. Журнал вошел в международные базы данных MathSciNet, zbMATH и список ВАК.

Кафедры и хронология руководства

Александр Константинович последовательно возглавлял несколько коллективов и структурных подразделений:

- 1989–2003 – заведующий кафедрой математического моделирования ОмГУ;
- 1989–2001 – заведующий совместной с ЦКБА лабораторией программного обеспечения и компьютерных сетей;
- 2003–2022 – заведующий кафедрой кибернетики ОмГУ;
- 2001–2022 – декан созданного им факультета компьютерных наук;
- 2008–2022 – директор Учебно-научного центра «Секрет-Информ» при факультете компьютерных наук ОмГУ.

Ученики и научная школа

Александр Константинович создал мощную научную школу. Под его руководством защитились 18 кандидатов наук и 1 доктор наук по специальностям: геометрия и топология, теоретическая физика, математическое моделирование и даже стоматология.

Роль в развитии ОмГУ

Омскому государственному университету Александр Константинович посвятил полвека жизни – с 1974 по 2024 год. Он прошёл путь от ассистента кафедры геометрии до профессора, заведующего кафедрами и декана. Именно благодаря его усилиям в университете сформировалась современная ИТ-школа, открылись новые специальности, была создана материальная база для подготовки специалистов по компьютерной безопасности и информационным технологиям.

Общий объём научного наследия Александра Константиновича впечатляет: более 500 научных работ и около 30 книг и учебников.

Политическая деятельность и взаимодействие с властью

В 1980-е годы Александр Константинович активно участвовал в общественно-политической жизни. Будучи членом КПСС с 1984 года, он выступал доверенным лицом известных политиков – А.И. Казанника (впоследствии Генерального прокурора РФ) и С.Н. Бабурина (заместителя Председателя Государственной Думы) на выборах народных депутатов СССР и РСФСР.

В 1992–1993 годах Гуц работал в Администрации Омской области в должности председателя информационно-аналитического комитета, а в 1993 году прошел повышение квалификации в Администрации Президента Российской Федерации. Этот опыт позволял ему эффективно выстраивать взаимодействие факультета с органами власти и силовыми структурами при подготовке специалистов по информационной безопасности.

Признание и награды

В 2017 году Александр Константинович был удостоен почетного звания «Заслуженный работник высшей школы Российской Федерации» за выдающиеся заслуги в области науки и образования.

Светлая память об Александре Константиновиче Гуце – выдающемся учёном, талантливом педагоге, основателе научной школы и факультета – навсегда сохранится в сердцах его коллег, учеников и всех, кому посчастливилось знать этого увлечённого наукой и преданного делу человека. Его идеи продолжают жить в трудах учеников, в созданных им структурах и на страницах журнала, который он вёл до последних дней.

Редакция выражает глубокие соболезнования родным и близким Александра Константиновича. Светлая память.

Редакция журнала

ОБ ОДНОМ ГИБРИДНОМ СПОСОБЕ РЕШЕНИЯ СЛОЖНОЙ СИСТЕМЫ ИЗ ДЕСЯТИ УРАВНЕНИЙ

Р.Н. Баротов¹

преподаватель, e-mail: ruzmet.tj@mail.ru

Д.Н. Баротов²

старший преподаватель, e-mail: DNBarotov@fa.ru

¹Худжандский государственный университет имени академика Б. Гафурова, Худжанд, Таджикистан

²Финансовый университет при Правительстве Российской Федерации, Москва, Россия

Аннотация. В данной работе изучается задача разрешимости системы из 10 уравнений, зависящей от целочисленного неотрицательного параметра n и описывающей особое состояние таблицы чисел, представляющее интерес, в том числе, с точки зрения математической игры. В результате изучения гибридным подходом устанавливается, что рассматриваемая система уравнений разрешима не при всех значениях параметра n , а именно, во-первых, сначала доказав первую априорную оценку решения, а затем на базе первой априорной оценки доказав более уточненную оценку и тем самым существенно сузив область поиска решения, а во-вторых, на основе улучшенной оценки перебрав на компьютере все оставшиеся варианты, которые могут быть решением, доказываемся неразрешимость рассматриваемой системы при $n = 6$. Также доказываемся критерий, когда $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ будет являться решением рассматриваемой системы при некотором значении параметра n . Помимо этого, в конце работы доказываемся факт, из которого, в частности, следует, что число значений параметра n , при которых рассматриваемая система разрешима, не мало, а именно, следует, что множество значений параметра n , при которых рассматриваемая система разрешима, счётно.

Ключевые слова: разрешимость системы, система уравнений, магическая таблица.

Краткое введение и постановка задачи

Изучение задачи, описывающей либо особое магическое состояние таблицы чисел, либо игру, связанную с таблицей чисел, является, с одной стороны, увлекательным занятием [1, 2], а с другой стороны, актуальным и важным. Потому что при изучении некоторых из таких задач, особенно при установлении неразрешимости задачи в специальном случае [2], когда в других случаях решение может быть найдено легко, например подбором, разрабатываются новые математические подходы, которые могут быть применены и обобщены при решении других аналогичных задач и построении и анализе математических игр [3].

В данной работе для произвольного неотрицательного целого числа n исследуем разрешимость системы из 10 уравнений вида

$$\begin{cases} x_0 = 1 + \beta_0(n) + \beta_0(x_0) + \beta_0(x_1) + \dots + \beta_0(x_9) \\ x_1 = 1 + \beta_1(n) + \beta_1(x_0) + \beta_1(x_1) + \dots + \beta_1(x_9) \\ \dots \\ x_9 = 1 + \beta_9(n) + \beta_9(x_0) + \beta_9(x_1) + \dots + \beta_9(x_9) \end{cases}, \quad (1)$$

где $\beta_k(n)$ – количество цифр $k \in \{0, 1, \dots, 9\}$ в неотрицательном целом числе n , а $x_0, x_1, \dots, x_9$ – искомые неизвестные натуральные числа, описывающей специальное магическое состояние таблицы чисел вида

	0	1	2	3	4	5	6	7	8	9
n	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9

(2)

а именно, для любой цифры $k \in \{0, 1, \dots, 9\}$ число x_k , стоящее под ней, равно количеству цифр k во всей таблице.

Отметим, что система (1) разрешима при $n \in \{0, 1, 2\}$. Действительно, для этого, ввиду вышеописанной связи между системой (1) и таблицей (2), достаточно привести следующие таблицы

0	0	1	2	3	4	5	6	7	8	9
	2	6	4	1	2	1	2	1	1	1

1	0	1	2	3	4	5	6	7	8	9
	1	8	3	2	1	1	1	1	2	1

2	0	1	2	3	4	5	6	7	8	9
	1	7	3	3	1	1	1	2	1	1

Основываясь на этой мотивации, изучаем разрешимость нелинейной системы из 10 диофантовых уравнений (1) для произвольного неотрицательного целого числа n . В результате изучения гибридным подходом устанавливается, что система (1) разрешима не при всех значениях параметра n , а именно, во-первых, сначала доказав первую априорную оценку решения, а затем на базе первой априорной оценки доказав более уточнённую оценку и тем самым существенно сузив область поиска решения, а во-вторых, на основе улучшенной оценки перебрав на компьютере все оставшиеся варианты, которые могут быть решением, доказываем неразрешимость системы (1) при $n = 6$. Также доказываем критерий, когда $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ будет являться решением системы (1) при некотором значении параметра n . Помимо этого, в конце работы доказываем факт, из которого, в частности, следует, что множество значений параметра n , при которых система (1) имеет не менее 3 решений, является счётным.

1. Используемые обозначения

Пусть $\alpha(n) = \begin{cases} 1 + [lg(n)], & \text{если } n \in \mathbb{N} \\ 1, & \text{если } n = 0 \end{cases}$ – количество всех цифр в неотрицательном целом числе n .

Пусть $\beta_k(n)$ – количество цифр k в неотрицательном целом числе n .

Пусть $\gamma(n) = \min_{k \in \{x \in \mathbb{N}: 10^{x-1} - 10 \cdot x - \alpha(n) \geq 1\}} k$ – минимальное натуральное решение неравенства $10^{x-1} - 10 \cdot x - \alpha(n) - 1 \geq 0$.

Замечание 1. Отметим, что для любого неотрицательного целого n положительное целое число $\gamma(n)$ определено корректно и также удовлетворяет неравенству

$$1 + lg(\alpha(n)) < \gamma(n) < 4 + lg(\alpha(n)).$$

2. Априорная оценка решения общей системы

Начнём изложение с формулировки и обоснования априорной оценки решения системы (1).

Утверждение 1. Если $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ является решением системы (1), то справедлива оценка

$$x_k^* < 10 \cdot \gamma(n) + \alpha(n) - 9 \quad \forall k \in \{0, 1, \dots, 9\}. \quad (3)$$

Доказательство. Прежде всего отметим, что $\exists i \in \{0, 1, \dots, 9\} : x_i^* \geq x_j^* \quad \forall j \in \{0, 1, \dots, 9\}$, иначе говоря, $x_i^* = \max(x_0^*, x_1^*, \dots, x_9^*)$. Пусть $x_i^* = \overline{a_{s-1}a_{s-2}\dots a_1a_0} - s$ -значное число, где $a_{s-1} \in \{1, \dots, 9\}$. Доказательство начнём с обоснования вспомогательной априорной оценки вида

$$\max(x_0^*, x_1^*, \dots, x_9^*) < 10^{\gamma(n)-1}. \quad (4)$$

От противного: пусть

$$x_i^* \geq 10^{\gamma(n)-1}. \quad (5)$$

Тогда, с одной стороны, ввиду $s, \gamma(n) \in \mathbb{N}$ получим, что имеет место цепочка

$$x_i^* \geq 10^{s-1} \geq 10^{\gamma(n)-1}, \quad (6)$$

а с другой стороны, ввиду $(i+1)$ -го уравнения системы (1), получим

$$x_i^* = 1 + \beta_i(n) + \beta_i(x_0^*) + \beta_i(x_1^*) + \dots + \beta_i(x_9^*) < 1 + \alpha(n) + 10 \cdot s. \quad (7)$$

Отсюда следует

$$10^{s-1} \leq x_i^* < 10 \cdot s + \alpha(n) + 1. \quad (8)$$

Но это противоречие, так как ввиду правой части (6) и определения $\gamma(n)$ имеем

$$10^{s-1} \geq 10 \cdot s + \alpha(n) + 1. \quad (9)$$

Справедливость вспомогательной априорной оценки вида (4) обоснована. Теперь, основываясь на (4), обоснуем более точную оценку вида (3). Действительно, в силу (4) имеем, что $x_i^* = \max(x_0^*, x_1^*, \dots, x_9^*)$ – не более чем $(\gamma(n) - 1)$ -значное число. Тогда в силу (7) имеем

$$x_i^* < 1 + \alpha(n) + 10 \cdot s \leq 10 \cdot (\gamma(n) - 1) + \alpha(n) + 1 = 10 \cdot \gamma(n) + \alpha(n) - 9. \quad (10)$$

Оценка, полученная в (10), завершает доказательство. Утверждение доказано. ■

3. Установление неразрешимости конкретной системы

В данном разделе докажем, что при $n = 6$ конкретная система уравнений вида (1) является несовместной.

Утверждение 2. Система уравнений вида

$$\begin{cases} x_0 = 1 + \beta_0(6) + \beta_0(x_0) + \beta_0(x_1) + \dots + \beta_0(x_9) \\ x_1 = 1 + \beta_1(6) + \beta_1(x_0) + \beta_1(x_1) + \dots + \beta_1(x_9) \\ \dots \\ x_9 = 1 + \beta_9(6) + \beta_9(x_0) + \beta_9(x_1) + \dots + \beta_9(x_9) \end{cases}, \quad (11)$$

где $(x_0, x_1, \dots, x_9) \in \mathbb{N}^{10}$, неразрешима.

Доказательство. Сначала докажем, что если $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ – есть решение системы (11), то имеет место следующая оценка

$$x_k^* \leq 9 \quad \forall k \in \{2, 3, 4, 5, 7, 8, 9\}. \quad (12)$$

Согласно вышеприведённому определению имеем, что $\gamma(n) = \gamma(6) = 3$. Следовательно, ввиду (3) имеем, что если $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ – есть решение системы (11), то справедливо неравенство

$$1 \leq x_j^* < 10 \cdot \gamma(6) + \alpha(6) - 9 = 22 \quad \forall j \in \{0, 1, \dots, 9\}. \quad (13)$$

На основании (13) справедливость (12) обоснуем от противного. Пусть $x_k^* \geq 10$ для некоторого $k \in \{2, 3, 4, 5, 7, 8, 9\}$. Тогда, ввиду $(k + 1)$ -го уравнения системы (11), оценки (13) и включения $k \in \{2, 3, 4, 5, 7, 8, 9\}$, получим

$$10 \leq x_k^* = 1 + \beta_k(6) + \beta_k(x_0^*) + \beta_k(x_1^*) + \dots + \beta_k(x_9^*) \leq 1 + 0 + 1 + 1 + \dots + 1 = 11. \quad (14)$$

Основываясь на (14) относительно возможного значения x_k^* , рассмотрим два случая.

Случай 1. Пусть $x_k^* = 11$. Тогда, ввиду $(k+1)$ -го уравнения системы (11), оценки (13) и включения $k \in \{2, 3, 4, 5, 7, 8, 9\}$, получим

$$11 = x_k^* = 1 + \beta_k(6) + \beta_k(11) + \sum_{\substack{j=0 \\ j \neq k}}^9 \beta_k(x_j^*) \leq 1 + 0 + 0 + \sum_{\substack{j=0 \\ j \neq k}}^9 1 = 10. \quad (15)$$

В (15) возникло противоречие.

Случай 2. Пусть $x_k^* = 10$. Тогда, ввиду $(k+1)$ -го уравнения системы (11), оценки (13) и включения $k \in \{2, 3, 4, 5, 7, 8, 9\}$, получим

$$x_j^* \in \{k, 10 + k\} \quad \forall j \in \{0, 1, \dots, 9\} \setminus \{k\}. \quad (16)$$

Основываясь на (16) и ввиду 1-го уравнения системы (11) получаем

$$\begin{cases} x_0^* \in \{k, 10 + k\} \\ x_0^* = 1 + \beta_0(6) + \beta_0(10) + \sum_{\substack{j=0 \\ j \neq k}}^9 \beta_0(x_j^*) = 2 \end{cases} \quad (17)$$

Если $k \neq 2$, то полученное в (17) противоречит $k \neq 2$, а если $k = 2$, то в 4-м уравнении системы (11) возникнет противоречие следующего вида

$$1 = 1 + \beta_3(6) + \beta_3(10) + \sum_{\substack{j=0 \\ j \neq k}}^9 \beta_3(x_j^*) = x_3^* \in \{2, 12\}. \quad (18)$$

Возникшее противоречие в (18) завершает обоснование справедливости оценки (12). Теперь, благодаря доказанной оценке (12), используя аналогичные рассуждения, нетрудно доказать, что если $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ – есть решение системы (11), то справедлива следующая оценка

$$x_k^* \leq 9 \quad \forall k \in \{0, 1, 6\}. \quad (19)$$

Объединяя полученные оценки (12) и (19), имеем

$$x_k^* \leq 9 \quad \forall k \in \{0, 1, \dots, 9\}. \quad (20)$$

Таким образом, с одной стороны, исходя из априорных оценок, имеем, что если система (11) имеет решение, то должно выполняться неравенство (20), а с другой стороны, перебрав на компьютере все варианты (20), которые могут быть решением, поскольку их осталось немного, нетрудно убедиться, что система (11) неразрешима. Утверждение доказано. ■

4. Мощность множества значений параметра n , при которых система разрешима

В этом разделе докажем, что множество значений параметра n , при которых система разрешима, является счётным.

Утверждение 3. Множество значений целочисленного неотрицательного параметра n , при которых система (1) разрешима, счётно.

Доказательство. Пусть S – множество значений целочисленного неотрицательного параметра n , при которых система (1) разрешима. Тогда имеет место включение

$$S \subset \{0, 1, 2, \dots\} = \{0\} \cup \mathbb{N}. \quad (21)$$

Теперь рассмотрим множество вида

$$S^* = \left\{ \underbrace{111\dots 11}_{\left(\frac{1}{9}(10^{m^*+2}-1)-m^*-12\right)\text{ раз}} : m \in \mathbb{N} \right\}. \quad (22)$$

Покажем, что имеет место включение

$$S^* \subset S. \quad (23)$$

Действительно, если $n = \underbrace{111\dots 11}_{\left(\frac{1}{9}(10^{m^*+2}-1)-m^*-12\right)\text{ раз}}$, то система (1) разрешима, так как нетрудно заметить, что

$$\left(1, \frac{1}{9}(10^{m^*+2}-1), 1, 1, 1, 1, 1, 1, 1, 1\right)$$

является решением системы (1). В силу счётности множеств S^* и $\{0\} \cup \mathbb{N}$ и включений (21) и (23) получаем, что множество S счётно. Утверждение доказано. ■

Теперь рассмотрим эту задачу с несколько иной позиции, а именно, выясним, когда конкретный упорядоченный набор натуральных чисел $(x_0^*, x_1^*, \dots, x_9^*)$ будет являться решением системы (1) при некотором n .

Утверждение 4. Для того чтобы упорядоченный набор натуральных чисел $(x_0^*, x_1^*, \dots, x_9^*)$ был решением системы (1) при некотором n , необходимо и достаточно, чтобы выполнялись следующие два условия:

- 1) для каждого $i \in \{0, 1, \dots, 9\}$ выполняется неравенство $m_i^* \geq 0$, где $m_i^* = x_i^* - 1 - \sum_{k=0}^9 \beta_i(x_k^*)$,
- 2) либо выполняется неравенство $(m_1^*, \dots, m_9^*) \neq (0, \dots, 0)$, либо выполняется равенство $(m_0^*, m_1^*, \dots, m_9^*) = (1, 0, \dots, 0)$.

Доказательство. Необходимость. Пусть $(x_0^*, x_1^*, \dots, x_9^*)$ является решением системы (1) при некотором $n^* \in \{0, 1, 2, \dots\}$. Тогда, ввиду определения $\beta_k(n)$ и $(i+1)$ -го уравнения системы (1), получаем

$$\beta_i(n^*) = x_i^* - 1 - \sum_{k=0}^9 \beta_i(x_k^*) = m_i^* \geq 0.$$

Чтобы обосновать справедливость условия 2), рассмотрим два случая.

Случай 1. Пусть $n^* = 0$. Тогда получим

$$(m_0^*, m_1^*, \dots, m_9^*) = (\beta_0(0), \beta_0(1), \dots, \beta_0(9)) = (1, 0, \dots, 0).$$

Случай 2. Пусть $n^* \in \mathbb{N}$. Тогда $\exists j \in \{1, 2, \dots, 9\}$ такой, что $\beta_j(n^*) \geq 1$ и, следовательно,

$$(m_1^*, \dots, m_9^*) = (\beta_1(n^*), \dots, \beta_9(n^*)) \neq (0, \dots, 0).$$

Достаточность. Пусть $(x_0^*, x_1^*, \dots, x_9^*)$ удовлетворяют условиям 1) и 2). Тогда, konstrуируя число в виде

$$n^* = \underbrace{999\dots 9}_{m_9^*-\text{раз}} \underbrace{888\dots 8}_{m_8^*-\text{раз}} \dots \underbrace{111\dots 1}_{m_1^*-\text{раз}} \underbrace{000\dots 0}_{m_0^*-\text{раз}}, \quad (24)$$

ввиду условий 1) и 2) число n^* корректно записано (определено), обоснуем, что $(x_0^*, x_1^*, \dots, x_9^*)$ будет решением системы (1) при $n = n^*$. Действительно, в силу (24), для каждого $i \in \{0, 1, \dots, 9\}$ имеем

$$x_i^* = 1 + \beta_i(n^*) + \sum_{k=0}^9 \beta_i(x_k^*) = 1 + m_i^* + \sum_{k=0}^9 \beta_i(x_k^*) = 1 + x_i^* - 1 = x_i^*.$$

Утверждение доказано. ■

В конце работы докажем еще одно утверждение, которое будет усилением утверждения 3.

Утверждение 5. Пусть $(s_{k,0}, s_{k,1}, \dots, s_{k,9})_{k=1}^m$ — m различных решений системы (1) при $n = n^* \in \mathbb{N}$ таких, что $s_{1,i} = s_{2,i} = \dots = s_{m,i} = s_i$ для некоторого $i \in \{0, 1, \dots, 9\}$. Тогда система (1) при $n = n^* \cdot 10^{9s_i+i-1} + \frac{i}{9}(10^{9s_i+i-1} - 1)$ имеет не менее m различных решений.

Доказательство. Достаточно обосновать, что $\forall k \in \{1, 2, \dots, m\}$ упорядоченный набор $(s_{k,0}, \dots, s_{k,i-1}, 10s_i + i, s_{k,i+1}, \dots, s_{k,9})$ является решением системы (1) при $n = n^* \cdot 10^{9s_i+i-1} + \frac{i}{9}(10^{9s_i+i-1} - 1)$. Действительно, если $j \in \{0, 1, \dots, 9\} \setminus \{i\}$, то ввиду того, что $(s_{k,0}, \dots, s_{k,i-1}, s_i, s_{k,i+1}, \dots, s_{k,9})$ является решением системы (1) при $n = n^*$, выполняется равенство

$$\begin{aligned} s_{k,j} &= 1 + \beta_j \left(n^* \cdot 10^{9s_i+i-1} + \frac{i}{9}(10^{9s_i+i-1} - 1) \right) + \beta_j(10s_i + i) + \sum_{\substack{l=0 \\ l \neq i}}^9 \beta_j(s_{k,l}) = \\ &= 1 + \beta_j(n^*) + \beta_j(s_i) + \sum_{\substack{l=0 \\ l \neq i}}^9 \beta_j(s_{k,l}) = 1 + \beta_j(n^*) + \sum_{l=0}^9 \beta_j(s_{k,l}) = s_{k,j}, \end{aligned} \quad (25)$$

а если $j = i$, то ввиду того, что $(s_{k,0}, \dots, s_{k,i-1}, s_i, s_{k,i+1}, \dots, s_{k,9})$ является решением системы (1) при $n = n^*$, выполняется равенство

$$\begin{aligned} 10s_i + i &= 1 + \beta_i \left(n^* \cdot 10^{9s_i+i-1} + \frac{i}{9}(10^{9s_i+i-1} - 1) \right) + \beta_i(10s_i + i) + \sum_{\substack{l=0 \\ l \neq i}}^9 \beta_i(s_{k,l}) = \\ &= 1 + \beta_i(n^*) + 9s_i + i - 1 + \beta_i(s_i) + 1 + \sum_{\substack{l=0 \\ l \neq i}}^9 \beta_i(s_{k,l}) = \\ &= 9s_i + i + \left(1 + \beta_i(n^*) + \sum_{l=0}^9 \beta_i(s_{k,l}) \right) = 9s_i + i + s_i = 10s_i + i. \end{aligned} \quad (26)$$

Ввиду произвольности $k \in \{1, 2, \dots, m\}$ в аргументированных равенствах (25) и (26), получим, что $(s_{k,0}, \dots, s_{k,i-1}, 10s_i + i, s_{k,i+1}, \dots, s_{k,9})$ является решением системы (1) при $n = n^* \cdot 10^{9s_i+i-1} + \frac{i}{9}(10^{9s_i+i-1} - 1)$ для любого $k \in \{1, 2, \dots, m\}$. Утверждение доказано. ■

Замечание 2. Отметим, что i -е координаты всех новых m различных решений, предъявленных в доказательстве утверждения 5, одинаковы и, следовательно, если система (1) имеет m различных решений с общей i -й, $i \in \{0, 1, \dots, 9\}$, координатой при $n = n^*$, то множество значений параметра n , при которых система (1) имеет не менее m различных решений с общей i -й координатой, является счётным.

Действительно, так как каждый раз, на основе текущего набора из m различных решений с общей i -й координатой при $n = n^*$, повторяя доказательство (процесс), предъявляем следующее натуральное число, которое больше n^* и такое, что система (1) имеет не менее m различных решений с общей i -й координатой при значении параметра n , равном которому.

Замечание 3. Объединение утверждения 5 и замечания 2, учитывая, что $(1, 6, 3, 3, 1, 1, 2, 1, 2, 1)$, $(1, 6, 4, 1, 2, 1, 2, 1, 2, 1)$ и $(1, 8, 1, 3, 1, 1, 1, 1, 3, 1)$ являются решениями системы (1) при $n = 8$, является усилением утверждения 3, а именно, влечёт справедливость, в частности, того факта, что множество значений параметра n , при которых система (1) имеет не менее 3 решений, является счётным.

Заключение

В работе благодаря доказанным априорным оценкам установлено, что система (1), зависящая от неотрицательного целого параметра n , разрешима не при всех значениях параметра n . Доказан критерий, когда $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ будет являться решением системы (1) при некотором n . Также доказано, что множество значений параметра n , при которых система (1) имеет не менее 3 решений, является счётным. Полученный результат и предложенный подход могут быть использованы при решении аналогичных задач, а также при построении и анализе математических компьютерных игр.

Литература

1. Кыдыралиев С.К., Урдалетова А.Б. Нестареющие магические квадраты. Часть 1 // Вестник Кыргызско-Российского Славянского университета. 2018. Т. 18, № 12. С. 15–19.
2. Долгов О.Т. Игра в 15 // Журнал «Квант». 1974. № 2. С. 26–33.
3. Гарднер М. Математические досуги. М.: Мир, 1972. 496 с.

ON A HYBRID METHOD FOR SOLVING A COMPLEX SYSTEM OF THE TEN EQUATIONS

R.N. Barotov¹

Instructor, e-mail: ruzmet.tj@mail.ru

D.N. Barotov²

Assistant Professor, e-mail: DNBarotov@fa.ru

¹Khujand State University named after Academician B. Gafurov, Khujand, Tajikistan

²Financial University under the Government of the Russian Federation, Moscow, Russia

Abstract. In this paper, we study the problem of solvability of a system of 10 equations depending on an integer non-negative parameter n and describing a special state of a table of numbers, which is of interest, among other things, from the point of view of a mathematical game. As a result of studying the hybrid approach, it is established that the system of equations under consideration is not solvable for all values of the parameter n , namely, first, by proving the first a priori estimate of the solution, and then, based on the first a priori estimate, proving a more refined estimate and thereby significantly narrowing the search area for a solution, and second, based on the improved estimate, having sorted out on a computer all the remaining options that can be a solution, the unsolvability of the system under consideration is proven for $n = 6$. We also prove a criterion when $(x_0^*, x_1^*, \dots, x_9^*) \in \mathbb{N}^{10}$ will be a solution to the system under consideration for a certain value of the parameter n . In addition, at the end of the work a fact is proved from which, in particular, it follows that the number of values of the parameter n for which the system under consideration is solvable is not small; namely, it follows that the set of values of the parameter n for which the system under consideration is solvable is countable.

Keywords: solvability of the system, system of equations, magic table.

Дата поступления в редакцию: 08.11.2024

ТЕОРЕТИКО-ГРУППОВОЕ ОПИСАНИЕ ПЕРИОДИЧЕСКОЙ СИСТЕМЫ ЭЛЕМЕНТОВ: VIII. СТРУКТУРА УРОВНЕЙ ЭНЕРГИИ

В.В. Варламов

д.ф.-м.н., профессор, e-mail: varlamov@subsiu.ru

Сибирский государственный индустриальный университет, Новокузнецк, Россия

Аннотация. Исследуется структура энергетических уровней периодической системы, состояния которой (химические элементы) представлены в виде узлов весовой диаграммы групповой алгебры группы вращений восьмимерного псевдоэвклидова пространства нейтральной сигнатуры.

Ключевые слова: таблица Менделеева, правило Маделунга, весовая диаграмма, вектор Лапласа–Рунге–Ленца, антивещество.

Введение

Как известно, схема Aufbau Бора [1] является исторически первой попыткой дать теоретическое объяснение строения периодической системы химических элементов. В основе этой схемы лежит модель атома Бора–Зоммерфельда, согласно которой атом имеет внутреннюю структуру, подобную солнечной системе в миниатюре (планетарная модель Резерфорда). Своеобразный симбиоз классической задачи Кеплера и квантовых постулатов¹ позволил Бору вывести формулу Ридберга и рассчитать энергетический спектр атома водорода (основные спектральные серии Лаймана, Бальмера и Пашена). Однако кроме спектра водорода² модель оказалась неспособной описать энергетические уровни многоэлектронных атомов (уже начиная с атома гелия).

Модель Бора считает, что электроны имеют известные орбиты и местоположения – две вещи, которые невозможно измерить одновременно согласно принципу неопределенности Гейзенберга. Электрону, как квантовому объекту, не присуще понятие классической траектории. Гейзенберг пишет: «... представление об электронной орбите, связанное с идеей дискретного стационарного состояния, было по ходу дела практически отброшено. Понятие дискретных стационарных состояний, однако, осталось жить. Понятие это было необходимо. Оно имело свою основу в

¹Что приводит к логической противоречивости, поскольку модель не является ни классической, ни квантовой: в системе двух уравнений, лежащих в её основе, одно – уравнение движения электрона – классическое, другое – уравнение квантования орбит – квантовое.

²Объяснение существования тонкой и сверхтонкой структуры спектральных линий водорода выходит за рамки описания модели Бора, как и описание эффекта Зеемана и аномальных ультрафиолетовых серий Пикеринга–Фаулера.

данных наблюдений. Наоборот, электронную орбиту не удалось согласовать с наблюдениями, поэтому от неё отказались, и от неё остались только матрицы для координат» [2, с. 97]. Дискретные стационарные состояния, соответствующие собственным значениям оператора энергии, – вот, что осталось от электронных орбит планетарной модели. Однако понятие электронной орбиты оказалось крайне живучим, трансформируясь в понятие атомной орбитали, т. е. в представление о том, что электрон в атоме имеет вид некоего электронного облака, различные формы которого определяются в зависимости от квантовых чисел n , l и m . Здесь следует отметить важный момент: все атомные орбитали описываются тремя квантовыми числами n , l и m , четвёртое квантовое число s (спин) в определении атомных орбиталей не участвует. Что касается реального существования атомных орбиталей, то говорить об их реальном существовании нет смысла. Это математическая абстракция, вопрос принципиально закрыт со времён Гейзенберга³.

Структура периодической системы, базирующейся на модели Бора (схеме Aufbau), исходит из того, что расположение элементов в системе при возрастании их атомных номеров однозначно определяется индивидуальными особенностями электронного строения атомов, описанного в рамках одноэлектронного приближения (метод Хартри), и непосредственно отражает энергетическую последовательность атомных орбиталей s -, p -, d -, f -оболочек, заселяющихся электронами при увеличении их суммарного числа по мере возрастания заряда ядра атома в соответствии с принципом минимума энергии. Однако это возможно лишь в самом простом варианте приближения Хартри, но уже в варианте приближения Хартри-Фока полная энергия атома не равна сумме орбитальных энергий, и электронная конфигурация атома определяется минимумом его полной энергии. Модель Бора не объясняет периодичность, а только лишь аппроксимирует её в рамках одноэлектронного приближения Хартри. Более того, Бор выводил электронные конфигурации не из квантовой теории, а исходя из известных химических и спектроскопических свойств элементов.

Нетрудно видеть, что в схеме Aufbau реализуется классический редукционистский принцип, согласно которому *целое* определяется *частями*, т. е. целое, понимаемое как *агрегат*, является производной от своих частей. *Холистической антитезой* к схеме Aufbau является модель Румера–Фета–Барута [7, 8], характерной чертой этой модели является представление периодической системы элементов как *единой квантовой системы*, состояния которой (химические элементы) соединяются друг с другом под действием группы симметрии. При этом под группой симметрии, как правило, понимаются тензорные расширения конформной группы: $G_F = \text{SO}(4, 2) \otimes \text{SU}(2) \otimes \text{SU}(2)'$ (группа Фета), $G_O = \text{O}(4, 2) \otimes \text{SU}(2)_S \otimes \text{SU}(2)_T$ (группа Островского). Групповой подход, в целом, представляет наиболее подходящую математическую структуру для описания явления периодичности, т. е. повторяемости (цикличности), что, как известно, есть главная характерная черта системы химических элементов.

Настоящая статья является продолжением серии работ [9–12], в которых реша-

³Однако в 1999 г. появляется статья [3] в престижном журнале «Nature», в которой говорилось о якобы экспериментально наблюдаемых атомных орбиталях. Возникла полемика [4, 5], в результате которой выяснилось, что авторы «открытия» на самом деле видели электронную плотность, а не атомные орбитали (более подробно см. [6]).

ется задача построения периодической системы химических элементов в рамках весовой диаграммы алгебры Ли четвёртого ранга группы вращений $SO(4, 4)$ восьмимерного псевдоэвклидова пространства $\mathbb{R}^{4,4}$ нейтральной сигнатуры. Исторически сложилось так, что графическая визуализация периодического закона имеет очень важное значение. Помимо первоначального 2-мерного табличного представления (таблицы Менделеева и Мейера) существует множество других различных форм: спиральные, винтовые и пирамидальные модели периодической системы, а также множество экзотических форм (более подробно см. [13–15]). Главной особенностью этих моделей является построение системы химических элементов в зависимости от возрастания атомного веса. Однако квантовая механика привела к пониманию, что главной структурной характеристикой периодического закона является не линейное возрастание атомного веса, а структура, определяемая порядком квантовых чисел. Polygonfläche Г. Генцеля [16] есть исторически первая модель такого рода, где структура периодической системы определяется порядком квантовых чисел, образующих системы концентрических многоугольников на плоскости. В. Финке в статье [17] представляет Polygonfläche Генцеля в трёхмерном пространстве, связав её с нумерацией Маделунга. С другой стороны, в подходе Румера–Фета–Барута структура «левосторонней» таблицы Жанэ связывается с квантовыми числами конформной группы $SO(4, 2)$ и её тензорного расширения G_F [7, 18], см. также [19–21]. Как известно, таблица Жанэ есть двумерное графическое представление правила Маделунга. Таким образом, посредством нумерации Маделунга имеется соответствие между моделью Генцеля–Финке и теоретико-групповым подходом Румера–Фета–Барута. Однако общим недостатком Polygonfläche Генцеля и трёхмерной системы Финке является искусственное представление четвёртого квантового числа s (спин) в виде двух точек на трансверсальных. В свою очередь, в теоретико-групповом подходе первые три квантовых числа n, l, m соответствуют собственным значениям генераторов Картана, образующих максимальную абелеву подалгебру алгебры Ли $\mathfrak{so}(4, 2)$ конформной группы. Алгебра $\mathfrak{so}(4, 2)$ имеет третий ранг. Чтобы включить четвёртое квантовое число s на равных основаниях с n, l, m , т. е. как собственное значение генератора Картана, требуется переход к алгебре Ли четвёртого ранга. В [11] показано, что такой алгеброй является $\mathfrak{so}(4, 4)$ – алгебра Ли группы $SO(4, 4)$. Соответствующий генератор спина L_{78} коммутирует со всеми генераторами подалгебры $\mathfrak{so}(4, 2)$, что приводит расщеплению («двуделению») базиса Картана–Вейля алгебры $\mathfrak{so}(4, 4)$. Как следствие, четырёхмерная весовая диаграмма алгебры $\mathfrak{so}(4, 4)$ может быть представлена двумя трёхмерными проекциями, каждая из которых изоморфна весовой диаграмме подалгебры $\mathfrak{so}(4, 2)$ [11, 12]. Следует отметить, что группа $SO(4, 4)$ обладает более высокой симметрией, чем тензорные расширения G_F и G_O : алгебра Ли $\mathfrak{so}(4, 4)$ содержит 28 независимых генераторов, в то время как алгебра Ли $\mathfrak{so}(4, 2) \otimes \mathfrak{su}(2) \otimes \mathfrak{su}(2)'$ групп G_F и G_O имеет 21 независимый генератор. Главным недостатком схемы Островского–Фета теоретико-группового описания удвоения периодов является искусственный характер введения пятого квантового числа, которое не имеет реального аналога, поскольку все состояния (элементы) периодической системы описываются четвёркой квантовых чисел (n, l, m, s) . Более того, введение пятого квантового числа s' приводит к изменению правила Маделунга (правило Фета [18]). В [12] показано, что четвёртый генератор Картана L_{78} (генера-

тор спина) в силу более высокой симметрии группы $SO(4, 4)$ объединяет («абсорбирует») функции генераторов Картана τ_3 и τ'_3 схемы Островского-Фета, что позволяет избежать введения дополнительного (нефизического) квантового числа s' . В п. 2 настоящей статьи две трёхмерные проекции весовой диаграммы алгебры $\mathfrak{so}(4, 4)$ объединяются в двойную $SO(4, 2)$ -башню, каждый узел которой (конечномерное представление группы $SO(4, 2)$) согласно нумерации Маделунга ассоциируется с соответствующим элементом периодической системы. Развёртка двойной $SO(4, 2)$ -башни на таблицу Жанэ и соответствие с моделью Румера-Фета-Барута устанавливается через базис Маделунга. Показывается, что основные энергетические уровни (этажи $SO(4, 2)$ -башни), определяемые собственными значениями радиального генератора $\mathbf{L}_{56}$, имеют структуру (j, j) -представлений Фока подгруппы $SO(4)$, где j – целое число. В п. 3 показано, что антивещество (антитаблица Менделеева, состоящая из антиэлементов: антиводород, антигелий, антилитий, ...) естественным образом включается в общую теоретико-групповую схему описания периодической системы.

1. Структура уровней

Геометрия периодической системы определяется порядковой структурой квантовых чисел. Согласно теоретико-групповому описанию периодической системы [11] первые три квантовых числа n , l и m соответствуют собственным числам ν , λ и μ_λ генераторов $\mathbf{L}_{56}$, $\mathbf{L}_{12}$ и $\mathbf{L}_{34}$, образующих подалгебру Картана $\mathfrak{K}$ алгебры Ли $\mathfrak{so}(4, 2)$ конформной группы $SO(4, 2)$. $\mathfrak{so}(4, 2)$ есть алгебра Ли третьего ранга, поэтому все корневые и весовые диаграммы этой алгебры являются трёхмерными системами. Для адекватного описания спина в рамках теоретико-групповой схемы требуется переход к алгебре Ли четвёртого ранга. Одной из таких алгебр является $\mathfrak{so}(4, 4)$ – алгебра Ли группы вращений $SO(4, 4)$ восьмимерного псевдоевклидова пространства $\mathbb{R}^{4,4}$ [11]. В этом случае подалгебра Картана $\mathfrak{K} \subset \mathfrak{so}(4, 4)$ содержит четыре генератора $\mathbf{L}_{56}$, $\mathbf{L}_{12}$, $\mathbf{L}_{34}$ и $\mathbf{L}_{78}$. Четвёртый генератор $\mathbf{L}_{78}$, понимаемый как генератор спина, коммутирует со всеми 15-ю генераторами подалгебры $\mathfrak{so}(4, 2)$. Как следствие, базис Картана–Вейля для алгебры $\mathfrak{so}(4, 4)$

$$\{\mathbf{L}_{12}, \mathbf{L}_{34}, \mathbf{L}_{56}, \mathbf{L}_{78}, {}^1\mathbf{K}_+, {}^1\mathbf{K}_-, {}^1\mathbf{J}_+, {}^1\mathbf{J}_-, {}^1\mathbf{T}_+, {}^1\mathbf{T}_-, {}^1\mathbf{S}_+, {}^1\mathbf{S}_-, {}^1\mathbf{P}_+, {}^1\mathbf{P}_-, {}^1\mathbf{Q}_+, {}^1\mathbf{Q}_-, {}^2\mathbf{K}_+, {}^2\mathbf{K}_-, {}^2\mathbf{J}_+, {}^2\mathbf{J}_-, {}^2\mathbf{T}_+, {}^2\mathbf{T}_-, {}^2\mathbf{S}_+, {}^2\mathbf{S}_-, {}^2\mathbf{P}_+, {}^2\mathbf{P}_-, {}^2\mathbf{Q}_+, {}^2\mathbf{Q}_-\}$$

расщепляется на два структурно идентичных базиса

$$\{{}^1\mathbf{K}_3, {}^1\mathbf{J}_3, {}^1\mathbf{T}_0, {}^1\mathbf{S}_0, {}^1\mathbf{P}_0, {}^1\mathbf{Q}_0, {}^1\mathbf{K}_+, {}^1\mathbf{K}_-, {}^1\mathbf{J}_+, {}^1\mathbf{J}_-, {}^1\mathbf{T}_+, {}^1\mathbf{T}_-, {}^1\mathbf{S}_+, {}^1\mathbf{S}_-, {}^1\mathbf{P}_+, {}^1\mathbf{P}_-, {}^1\mathbf{Q}_+, {}^1\mathbf{Q}_-\}, \quad (1)$$

$$\{{}^2\mathbf{K}_3, {}^2\mathbf{J}_3, {}^2\mathbf{T}_0, {}^2\mathbf{S}_0, {}^2\mathbf{P}_0, {}^2\mathbf{Q}_0, {}^2\mathbf{K}_+, {}^2\mathbf{K}_-, {}^2\mathbf{J}_+, {}^2\mathbf{J}_-, {}^2\mathbf{T}_+, {}^2\mathbf{T}_-, {}^2\mathbf{S}_+, {}^2\mathbf{S}_-, {}^2\mathbf{P}_+, {}^2\mathbf{P}_-, {}^2\mathbf{Q}_+, {}^2\mathbf{Q}_-\}, \quad (2)$$

каждый из которых приводит к весовой диаграмме подалгебры $\mathfrak{so}(4, 2)$. На рис. 1

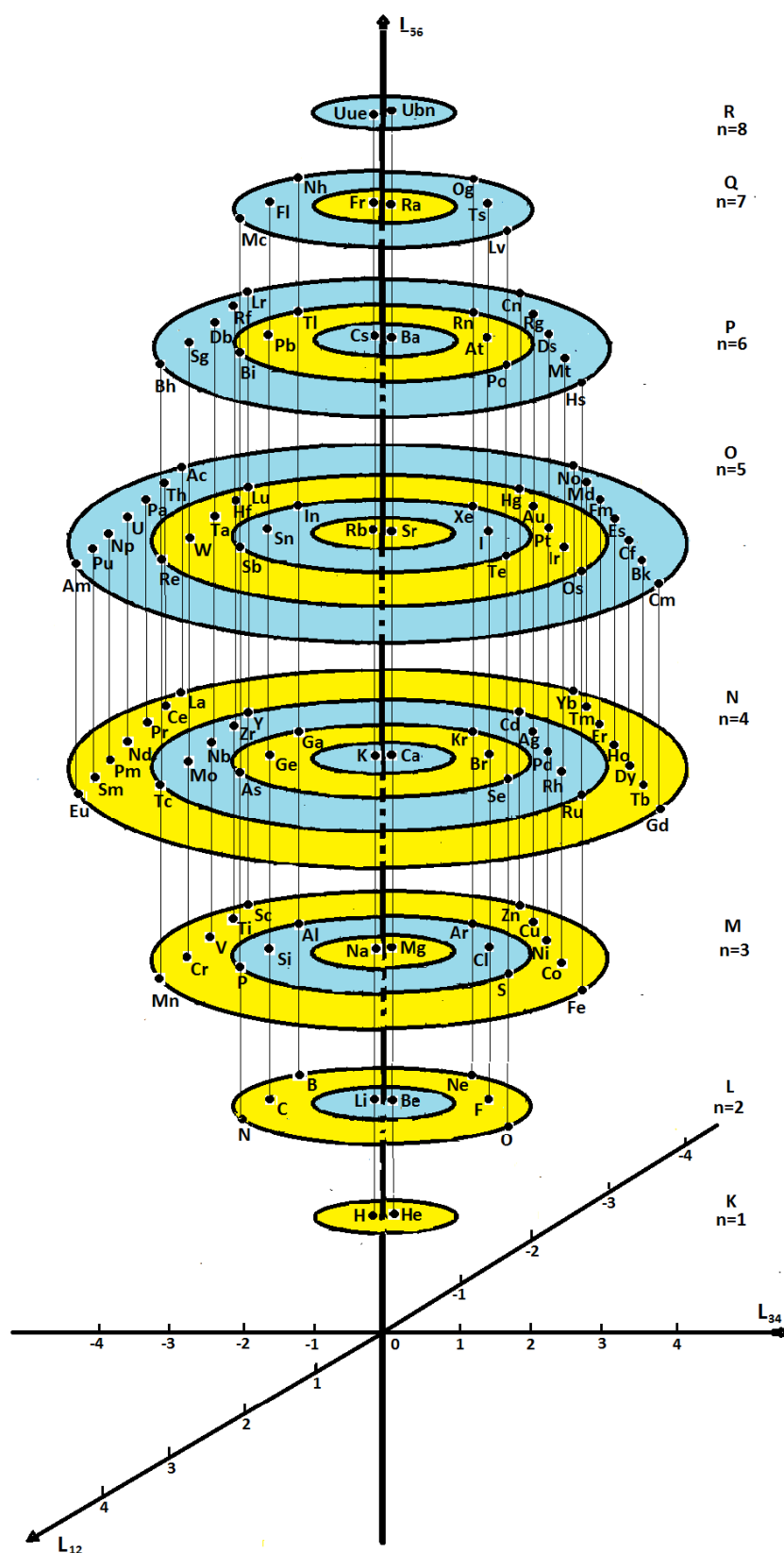


Рис. 1. Периодическая система химических элементов в форме совмещённой весовой диаграммы расщеплённых базисов (1) и (2) алгебры Ли $so(4, 4)$

представлены две совмещённые трёхмерные проекции ($SO(4, 2)$ -башни) весовой диаграммы алгебры Ли $\mathfrak{so}(4, 4)$ группы вращений $SO(4, 4)$, соответствующие базисам (1) и (2). Вертикальная ось $SO(4, 2)$ -башни образована собственными значениями генератора Картана $\Delta_3 = \mathbf{L}_{56}$, что добавляет к многообразию радиальный лестничный оператор. Каждый заданный этаж $SO(4, 2)$ -башни характеризуется главным квантовым числом n . Горизонтальные полосы (на этажах) соответствуют различным l -подоболочкам, а точки являются индивидуальными m -компонентами (конечномерными представлениями группы $SO(4, 2)$), задающими элементы периодической системы согласно правилу Маделунга. Удвоение периодов представлено следующим образом: жёлтым цветом обозначены кольца, содержащие элементы с нечётной суммой $n + l$, соответственно голубым цветом обозначены кольца с чётной суммой $n + l$. Гомологичные элементы соединяются вертикальными линиями. На радиальной оси слева ($s = -1/2$) расположены щелочные металлы I группы (**Li**, **Na**, ...), соответственно справа ($s = +1/2$) радиальную ось населяют щелочноземельные металлы II группы (**Be**, **Mg**, ...). По мере удаления от радиальной оси к перифериям кругов Генцеля происходит уменьшение *металличности* элементов и нарастание *неметалличности*, этому соответствует движение слева направо по периоду в стандартной таблице Менделеева (металлы $\rightarrow$ амфотерные элементы $\rightarrow$ инертные газы).

Легко видеть, что на каждом этаже двойной $SO(4, 2)$ -башни, показанной на рис. 1, реализуется структура (j, j) -представлений Фока подгруппы $SO(4)$ (целочисленная часть весовой диаграммы подалгебры $\mathfrak{so}(4)$, см. рис. 3 в [10]). Первый этаж ($n = 1$) содержит первый период, состоящий из водорода **H** и гелия **He**. **H** и **He** образуют двойной синглет. На этаже $n = 2$ m -компоненты (элементы периодической системы) образуют $(1, 1)$ -диаграмму подалгебры $\mathfrak{so}(4)$ (см. рис. 2), содержащей второй период: двойной синглет (**Be**, **Li**) и два триплета (**B**, **C**, **N**), (**Ne**, **F**, **O**). Далее, на листе $n = 3$ (см. рис. 3) имеем $(2, 2)$ -диаграмму подалгебры $\mathfrak{so}(4)$, содержащей 3-й период полностью, а также переходные металлы побочных подгрупп 4-го периода от скандия **Sc** до марганца **Mn** и от железа **Fe** до цинка **Zn**, образующие два квинтета. Поднимаясь выше на следующий $n = 4$ этаж $SO(4, 2)$ -башни, приходим к $(3, 3)$ -диаграмме подалгебры $SO(4)$, на котором завершается заполнение 4-го периода инертным газом **Kr** (криптон), см. рис. 4. $(3, 3)$ -диаграмма на рис. 4 также содержит переходные металлы 5-го периода от иттрия **Y** до технеция **Tc** и от рутения **Ru** до кадмия **Cd**. Здесь же, в рамках $(3, 3)$ -диаграммы рис. 4, расположено всё семейство лантаноидов **La**, **Ce**, ..., **Yb**. Поднимаясь ещё выше на 5-й этаж, приходим к ещё одной $(3, 3)$ -диаграмме подалгебры $\mathfrak{so}(4)$, см. рис. 5, на которой, в свою очередь, завершается заполнение 5-го периода инертным газом **Xe** (ксенон). $(3, 3)$ -диаграмма на рис. 5 содержит переходные металлы 6-го периода от лютеция **Lu** до рения **Re** и от осмия **Os** до ртути **Hg**. Здесь же, в рамках $(3, 3)$ -диаграммы рис. 5, находится всё семейство актиноидов **Ac**, **Th**, ..., **No**. Далее, на листе $n = 6$ (см. рис. 6) имеем $(2, 2)$ -диаграмму подалгебры $\mathfrak{so}(4)$, на которой завершается заполнение 6-го периода инертным газом **Rn** (радон). В рамках этой диаграммы содержатся переходные металлы 7-го периода от лоуренсия **Lr** до бория **Bh** и от хассия **Hs** до коперниция **Cn**. Наконец, на этаже $n = 7$ (см. рис. 7) имеем $(1, 1)$ -диаграмму подалгебры $\mathfrak{so}(4)$, на которой завершается заполнение 7-го периода инертным газом **Og** (оганесон).

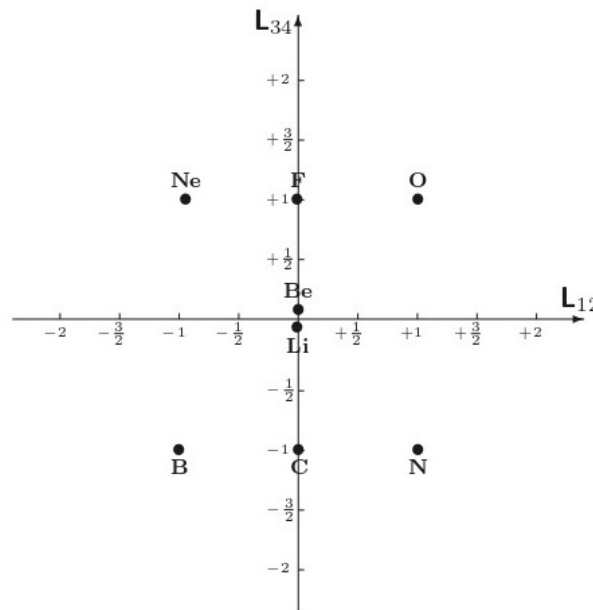


Рис. 2. Уровень $n = 2$ весовой диаграммы алгебры $\mathfrak{so}(4, 4)$ содержит $(1, 1)$ -диаграмму подалгебры $\mathfrak{so}(4)$, полностью включающей в себя второй период

Седьмой период завершает заполнение таблицы Менделеева. Этаж $n = 8$ содержит двойной синглет гипотетических элементов **Uue** и **Ubn**, с которых начинается построение таблицы Сиборга и далее 10-периодического расширения.

Таким образом, все элементы таблицы Менделеева группируются в последовательность весовых $SO(4)$ -диаграмм подалгебры $\mathfrak{so}(4)$, расположенной на восьми этажах весовой диаграммы (двойной $SO(4, 2)$ -башни, рис. 1) алгебры $\mathfrak{so}(4, 4)$.

С целью вычислить энергетические (массовые) уровни единой квантовой системы, определяемой $SO(4)$ -мультиплетами, образуем сумму и разность двух операторов Казимира $\mathbf{K}^2$ и $\mathbf{J}^2$ группы $SO(4)$:

$$\begin{aligned}\mathbf{K}^2 + \mathbf{J}^2 &= \frac{1}{4}(\mathbf{L} + \mathbf{A})^2 + \frac{1}{4}(\mathbf{L} - \mathbf{A})^2 \\ &= \frac{1}{4}(\mathbf{L}^2 + \mathbf{A}^2 + \mathbf{L} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{L}) + \frac{1}{4}(\mathbf{L}^2 + \mathbf{A}^2 - \mathbf{L} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{L}) \quad (3) \\ &= \frac{1}{2}(\mathbf{L}^2 + \mathbf{A}^2),\end{aligned}$$

$$\begin{aligned}\mathbf{K}^2 - \mathbf{J}^2 &= \frac{1}{4}(\mathbf{L} + \mathbf{A})^2 - \frac{1}{4}(\mathbf{L} - \mathbf{A})^2 \\ &= \frac{1}{4}(\mathbf{L}^2 + \mathbf{A}^2 + \mathbf{L} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{L}) - \frac{1}{4}(\mathbf{L}^2 + \mathbf{A}^2 - \mathbf{L} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{L}) \quad (4) \\ &= \frac{1}{2}(\mathbf{L} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{L}).\end{aligned}$$

В силу ортогональности вектора орбитального момента $\mathbf{L}$ и вектора Лапласа-Рунге-

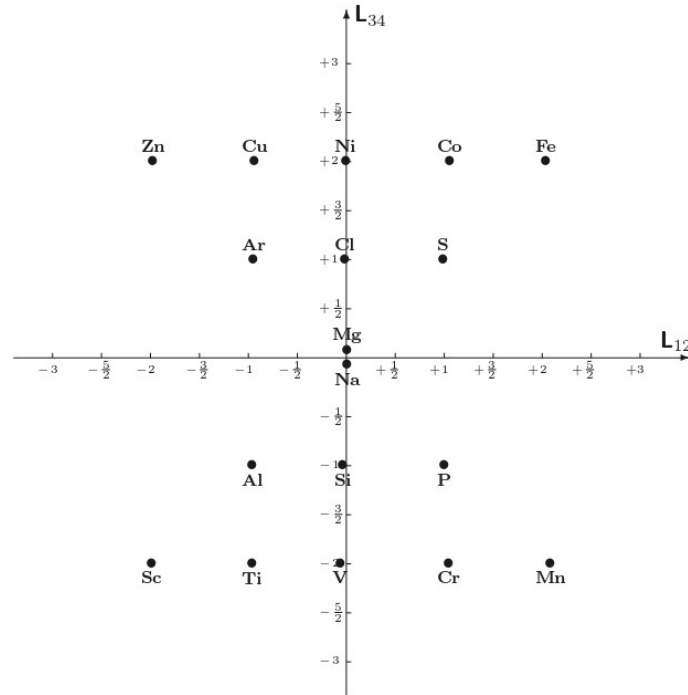


Рис. 3. Уровень $n = 3$: $(2, 2)$ -диаграмма подалгебры $\mathfrak{so}(4)$ содержит полностью 3-й период и переходные металлы 4-го периода **Sc**, ..., **Mn**, **Fe**, ..., **Zn**

Ленца $\mathbf{M}$, $\mathbf{L} \cdot \mathbf{M} = 0$, и квантовомеханического определения оператора

$$A = \frac{\hat{\mathbf{M}}}{\sqrt{-2mE}} \quad (5)$$

скалярные произведения $\mathbf{L} \cdot \mathbf{A}$ и $\mathbf{A} \cdot \mathbf{L}$ исчезают, что приводит (4) к виду

$$\mathbf{K}^2 - \mathbf{J}^2 = 0, \quad \mapsto \quad \mathbf{K}^2 = \mathbf{J}^2. \quad (6)$$

Классический вектор Лапласа–Рунге–Ленца (ЛРЛ) описывает *прецессию эллиптической орбиты* в задаче Кеплера. В рамках планетарной модели Резерфорда–Бора ЛРЛ-вектор для атома водорода определяется следующим образом:

$$\mathbf{M} = \mathbf{p} \times \mathbf{L} - mk \frac{\mathbf{r}}{r}, \quad (7)$$

где r – расстояние между протоном и электроном, $m = m_p m_e / (m_p + m_e)$ – приведённая масса системы протон+электрон, $\mathbf{p}$ и $\mathbf{L}$ – импульс и орбитальный момент системы, $k = Ze^2 / (4\pi\epsilon_0)$. Поскольку ЛРЛ-вектор $\mathbf{M}$ лежит в плоскости орбиты (и касателен к ней), то он ортогонален орбитальному моменту $\mathbf{L}$, следовательно

$$\mathbf{L} \cdot \mathbf{M} = 0, \quad (8)$$

$$\mathbf{M}^2 = 2m\mathcal{H}\mathbf{L}^2 + m^2 k^2, \quad (9)$$

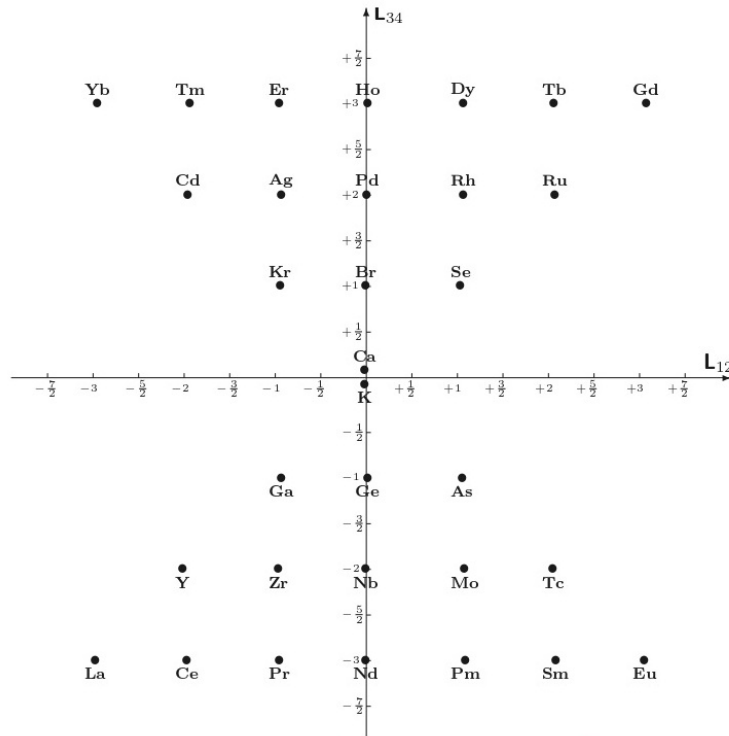


Рис. 4. Уровень $n = 4$: $(3, 3)$ -диаграмма подалгебры $\mathfrak{so}(4)$ содержит завершение 4-го периода, переходные металлы 5-го периода **Y**, ..., **Tc**, **Ru**, ..., **Cd**, а также семейство лантаноидов **La**, **Ce**, ..., **Yb**

где

$$\mathcal{H} = \frac{p^2}{2m} - \frac{k^2}{r}$$

– гамильтониан системы.

Отсюда следует, что уровни энергии квантовой системы характеризуются равенством квантовых чисел (собственных значений генераторов Картана $\mathbf{K}_3$ и $\mathbf{J}_3$) группы $SO(4)$:

$$j_1 = j_2. \quad (10)$$

Условие (10) определяет представление Фока, рассмотренное в приложении А статьи [23], где это равенство получено другим способом. Таким образом, m -компоненты (элементы таблицы Менделеева) группируются на этажах двойной $SO(4, 2)$ -башни в последовательность весовых (j, j) -диаграмм подалгебры $SO(4)$, где j – целое число.

Учитывая (6), перепишем (3) следующим образом:

$$2\mathbf{K}^2 = \frac{1}{2}(\mathbf{L}^2 + \mathbf{A}^2) \quad \longmapsto \quad 4\mathbf{K}^2 = \mathbf{L}^2 + \mathbf{A}^2. \quad (11)$$

Тогда левая часть равенства (11) с учётом собственных значений оператора Казимира $\mathbf{K}^2$ запишется

$$4\mathbf{K}^2 = 4j(j+1)\hbar^2, \quad \text{где} \quad j = 0, \frac{1}{2}, 1, \frac{3}{2}, \dots \quad (12)$$

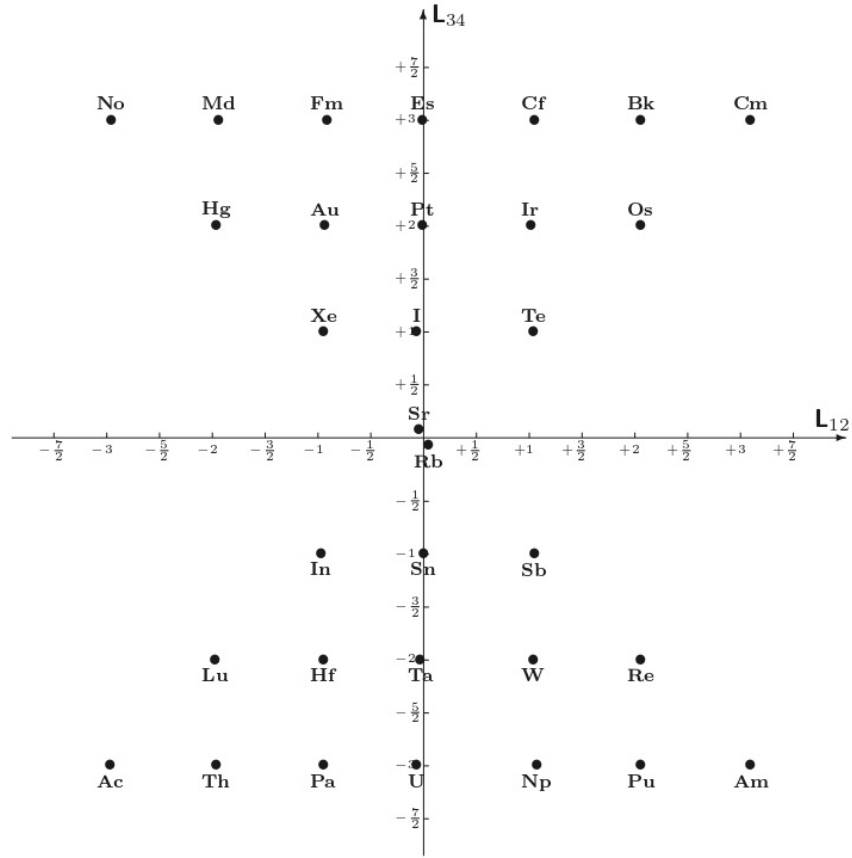


Рис. 5. Уровень $n = 5$: $(3, 3)$ -диаграмма подалгебры $\mathfrak{so}(4)$ содержит завершение 5-го периода, переходные металлы 6-го периода **Lu**, ..., **Re**, **Os**, ..., **Hg**, а также семейство актиноидов **Ac**, **Th**, ..., **No**

Записывая квантовомеханический аналог формулы (9) в виде

$$\frac{\mathbf{M}^2}{2mE} = \mathbf{L}^2 + \hbar^2 + \frac{mk^2}{2E},$$

получим с учётом (5)

$$\mathbf{L}^2 + \mathbf{A}^2 = \mathbf{L}^2 - \frac{\mathbf{M}^2}{2mE} = -\frac{mk^2}{2E} - \hbar^2.$$

Комбинируя (11) с (12), находим

$$4j(j+1)\hbar^2 = -\frac{mk^2}{2E} - \hbar^2.$$

Откуда следует

$$E_j = -\frac{mk^2}{2\hbar^2(2j+1)^2}, \quad j = 0, \frac{1}{2}, 1, \frac{3}{2}, \dots \quad (13)$$

Учитывая, что $k = Ze^2/4\pi\epsilon_0$, $\hbar = h/2\pi$ и $n = 2j + 1$, получим

$$E_n = -\frac{m_e Z^2 e^4}{8\hbar^2 \epsilon_0^2 n^2}, \quad n = 1, 2, 3, \dots \quad (14)$$

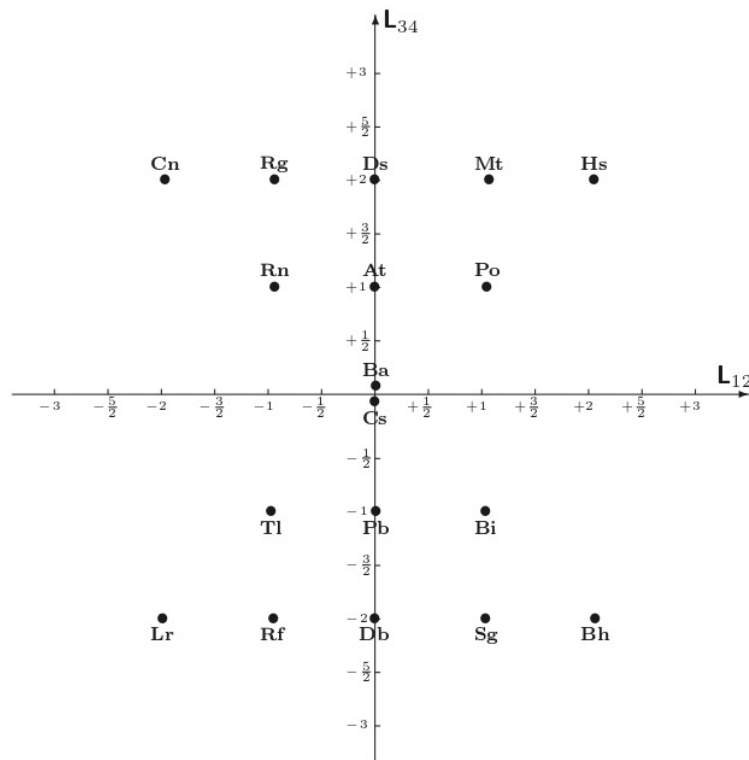


Рис. 6. Уровень $n = 6$: $(2, 2)$ -диаграмма подалгебры $\mathfrak{so}(4)$ содержит завершение 6-го периода и переходные металлы 7-го периода **Lr**, ..., **Bh**, **Hs**, ..., **Cn**

Принимая во внимание связь массы с энергией, $E = mc^2$, из последнего соотношения получим массы состояний единой квантовой системы:

$$m_n = \frac{m_e Z^2 e^4}{8h^2 \varepsilon_0^2 c^2} \cdot \frac{1}{n^2}, \quad n = 1, 2, 3, \dots \quad (15)$$

Формула (15) учитывает только главное квантовое число (при $l = 0$), соответствующее собственным значениям радиального генератора $\mathbf{L}_{56} = \Delta_3$. Следовательно, формула (15) задаёт массы состояний, лежащих на радиальной оси двойной $SO(4, 2)$ -башни (весовая диаграмма на рис. 1), т. е. массы щелочных и щелочноземельных металлов.

На каждом этаже левой ($s = -1/2$) и правой ($s = +1/2$) башен (см. рис. 3 в [12]) находится n^2 состояний, что соответствует размерности представлений подгруппы $SO(4)$, а на этажах совмещённой башни (см. рис. 1) имеем реализации весовых диаграмм алгебры $\mathfrak{so}(4)$. Таким образом, число состояний (элементов) на каждом этаже совмещённой $SO(4, 2)$ -башни определяется последовательностью Ридберга $2n^2$:

$$2 = 2 \cdot 1^2, \quad 8 = 2 \cdot 2^2, \quad 18 = 2 \cdot 3^2, \quad 32 = 2 \cdot 4^2, \quad \dots, \quad (16)$$

которую Зоммерфельд называл «кабалистической» в своей книге [24, С. 122]. Далее имеет место удвоение периодов (за исключением первого):

$$2, \quad 8, \quad 8, \quad 18, \quad 18, \quad 32, \quad 32, \quad \dots \quad (17)$$

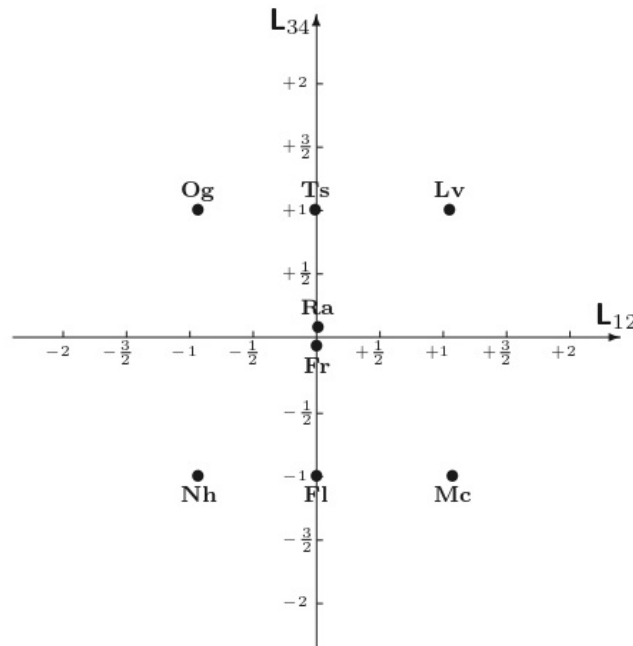


Рис. 7. Уровень $n = 7$: $(1, 1)$ -диаграмма подалгебры $\mathfrak{so}(4)$ завершает таблицу Менделеева заполнением 7-го периода до инертного газа **Og** (оганессон)

Первое удвоение (16) иногда называют «горизонтальным» (или *спиновым*), второе (17) – «вертикальным». В схеме Островского-Фета первое удвоение реализуется с помощью генератора Картана τ_3 с соответствующим квантовым числом s (собственным значением), удвоение (17) вводится посредством второго генератора Картана τ'_3 с соответствующим пятым (нефизическим) квантовым числом s' , приводящим к изменению правила Маделунга.

В рамках предложенной схемы описания периодической системы четвёртый генератор L_{78} подалгебры Картана алгебры $\mathfrak{so}(4, 4)$ объединяет функции обоих генераторов τ_3 и τ'_3 схемы Островского-Фета в силу более высокой симметрии группы $SO(4, 4)$: алгебра Ли $\mathfrak{so}(4, 4)$ содержит 28 независимых генераторов, в то время как групповая алгебра $\mathfrak{so}(4, 2) \otimes \mathfrak{su}(2) \otimes \mathfrak{su}(2)'$ тензорного расширения G_F имеет 21 независимый генератор. Коммутация генератора L_{78} со всеми 15-ю генераторами подалгебры $\mathfrak{so}(4, 2)$ приводит к расщеплению базиса Картана–Вейля алгебры $\mathfrak{so}(4, 4)$ (см. [11]) и, соответственно, к первому удвоению (16). Весовая диаграмма алгебры $\mathfrak{so}(4, 4)$ четырёхмерна, ось генератора L_{78} перпендикулярна к оси радиального генератора L_{56} , а также к осям генераторов L_{12} и L_{34} , т. е. ось генератора L_{78} перпендикулярна к трёхмерному весовому пространству, базисом которого служат оси генераторов L_{12} , L_{34} , L_{56} , что приводит, в свою очередь, к «вертикальному» (фактическому) удвоению периодов (17). При этом исчезает необходимость введения пятого квантового числа.

Таким образом, *фактическое удвоение периодов (17) есть результат действия четвёртого генератора L_{78} (генератор спина) подалгебры Картана $\mathfrak{K}$ алгебры Ли*

$\mathfrak{so}(4, 4)$. При этом спин есть четвёртая степень свободы, не имеющая аналога в рамках классической трёхмерной системы. *Спин – это эффект четвёртого измерения*⁴.

2. Включение антивещества

Первые свидетельства существования *антиводорода* были получены на ускорителях в 1990-х годах. Впервые захват атомов антиводорода в ловушку продемонстрировала группа Antihydrogen Laser Physics Apparatus (ALPHA) в ЦЕРН в 2010 году. Логично предположить, что существование атомов антиводорода влечёт за собой существование атомов *антигелия*, *антилития* и т. д., т. е. существование двойственной *антипериодической таблицы Менделеева*. Теоретико-групповое описание естественным образом включает в рассмотрение *антиэлементы* периодической системы уже на уровне подгруппы $SO(4, 2)$. Весовая диаграмма алгебры $\mathfrak{so}(4, 2)$ состоит из двух пирамид, симметричных относительно плоскости, образованной генераторами Картана L_3 и A_3 (см. рис. 6 в [10]). При этом собственное число радиального генератора Δ_3 для нижней пирамиды принимает отрицательные значения. Таким образом, включение антивещества в общую теоретико-групповую схему описания периодической системы требует введения отрицательных значений главного квантового числа n , т. е. собственных значений радиального генератора L_{56} . Ограничение только положительными значениями числа n в модели Бора обусловлено геометрическими характеристиками кеплеровских орбит планетарной модели Резерфорда⁵ (более подробно см. [24]).

В свою очередь, для алгебры $\mathfrak{so}(4, 4)$ весовое пространство определяется четырьмя генераторами Картана $L_{56} = \Delta_3$, $L_{12} = L_3$, $L_{34} = A_3$ и L_{78} , которые служат

⁴Тот вещественный трёхмерный образ атома, формируемый средствами регистрирующей аппаратуры (масс-спектрометры, электронные и туннельные микроскопы) есть не более чем верхушка айсберга, основная часть которого погружена в мир четырёх (а может быть и более) измерений. Гейзенберг говорил: «Атомы – это не вещи».

⁵Следует отметить, что в модели Бора квантовые числа n, l, m соответствуют радиальным и орбитальным движениям классической системы (задачи Кеплера), описывающим движения «составных частей» атома. При этом четвёртое квантовое число s не имеет классического аналога, что лишний раз указывает на паллиативный характер этой модели. А. Барут в статье [8] о групповой структуре периодической системы в рамках конформной группы $SO(4, 2)$ вводит квантовые числа $\nu, \lambda, \mu_\lambda$ группы $SO(4, 2)$ с целью рассмотрения химических элементов как различных *состояний* единой *квантовой системы*. Чтобы не быть предвзятым по отношению к квантовым числам n, l, m механической (планетарной) модели Бора, описывающей водородоподобные системы, Барут намеренно вводит символы $\nu, \lambda, \mu_\lambda$, имеющие, прежде всего, теоретико-групповой (не механический) смысл, хотя диапазон их изменения Барут оставляет таким же, как у «водородных» квантовых чисел модели Бора. Очевидно, что это ограничение диапазона изменения чисел $\nu, \lambda, \mu_\lambda$, являющихся собственными числами генераторов Картана алгебры $\mathfrak{so}(4, 2)$, является искусственным допущением, следующим из требования согласия с моделью Бора. Островский аналогичным образом проводит различие между обычными («водородными») квантовыми числами и абстрактными $SO(4, 2)$ -символами, обозначая последние знаком тильды: $\tilde{n}, \tilde{l}, \tilde{m}_l$ [22]. Фет в книге [18] придерживается символизма Барута. Следуя Баруту, Островский и Фет ограничивают диапазон изменения собственного числа ν радиального генератора положительными значениями.

базисом четырёхмерной ортогональной системы координат. Определим кет-вектор

$$\left| \nu, \dot{\nu}; \lambda, \dot{\lambda}; \mu, \dot{\mu}; \sigma, \dot{\sigma} \right\rangle, \quad (18)$$

где

$$\begin{aligned} \nu, \dot{\nu} &\in \{0, 1/2, 1, 3/2, \dots\}, \\ \lambda &\in \{-\nu, -\nu + 1, \dots, \nu - 1, \nu\}, \quad \dot{\lambda} \in \{-\dot{\nu}, -\dot{\nu} + 1, \dots, \dot{\nu} - 1, \dot{\nu}\}, \\ \mu &\in \{-\lambda, -\lambda + 1, \dots, \lambda - 1, \lambda\}, \quad \dot{\mu} \in \{-\dot{\lambda}, -\dot{\lambda} + 1, \dots, \dot{\lambda} - 1, \dot{\lambda}\}, \\ \sigma, \dot{\sigma} &\in \{-1/2, +1/2\}. \end{aligned}$$

Связь чисел, входящих в кет-вектор (18), с квантовыми числами n, l, m и s кет-вектора

$$|n, l, m, s\rangle \quad (19)$$

базиса Маделунга (см. табл. 1 в [12]) задаётся следующими соотношениями

$$n = |\nu - \dot{\nu}|, \quad l = |\lambda - \dot{\lambda}|, \quad m = |\mu - \dot{\mu}|, \quad s = |\sigma - \dot{\sigma}|.$$

При этом главное квантовое число n изменяется в пределах

$$-|\nu - \dot{\nu}|, -|\nu - \dot{\nu}| + 1, -|\nu - \dot{\nu}| + 2, \dots, |\nu - \dot{\nu}|. \quad (20)$$

На рис. 8 показана совмещённая диаграмма двух трёхмерных проекций весовой диаграммы алгебры $\mathfrak{so}(4, 4)$. Верхняя башня, начинающаяся с водорода $\mathbf{H}$, представляет *пирамиду вещества*, где собственные числа радиального генератора $\mathbf{L}_{56} = \Delta_3$ принимают положительные значения. Нижняя башня, отражённая от плоскости, образованной осями генераторов $\mathbf{L}_{12} = \mathbf{L}_3$ и $\mathbf{L}_{34} = \mathbf{A}_3$, начиная с антиводорода $\overline{\mathbf{H}}$, представляет *пирамиду антивещества*, где собственные числа генератора Δ_3 принимают отрицательные значения.

Литература

1. Бор Н. Избранные научные труды. Т. 1. М.: Наука, 1970. С. 318–375.
2. Гейзенберг В. Шаги за горизонт. М.: Прогресс, 1987.
3. Zuo J.M., Kim M., O’Keeffe M., Spence J.C.H. // Nature. 1999. V. 401. P. 49–52.
4. Scerri E.R. // J. Chem. Educ. 2000. V. 77. P. 1492.
5. Spence J.C.H., O’Keeffe M., Zuo J.M. // J. Chem. Educ. 2001. V. 78. P. 877.
6. Варламов В.В. Существуют ли атомные орбитали? URL: <https://zen.ru/a/Z1fteLQ63GGu9y5G> (дата обращения 21.12.2025).
7. Румер Ю.Б., Фет А.И. Группа $\text{Spin}(4)$ и таблица Менделеева // Теоретическая и математическая физика. 1971. Т. 9. С. 203–209.
8. Barut A.O. Group Structure of the Periodic System // The Structure of Matter: Rutherford Centennial Symposium / ed. by B.G. Wybourne. Christchurch, New Zeland: University of Canterbury Press, 1972. P. 126–136.

9. Варламов В.В. Теоретико-групповое описание периодической системы элементов. IV: Групповая алгебра // Математические структуры и моделирование. 2024. № 1 (69). С. 18–31.
10. Варламов В.В. Теоретико-групповое описание периодической системы элементов. V: Весовая диаграмма // Математические структуры и моделирование. 2024. № 3 (71). С. 4–18.
11. Варламов В.В. Теоретико-групповое описание периодической системы элементов. VI: Восьмимерное обобщение // Математические структуры и моделирование. 2025. № 2 (74). С. 18–32.
12. Варламов В.В. Теоретико-групповое описание периодической системы элементов. VII: Спин и удвоение периодов // Математические структуры и моделирование. 2025. № 3 (75). С. 26–41.
13. van Spronsen J.W. The Periodic System of Chemical Elements. Amsterdam / New York: Elsevier, 1969.
14. Mazurs E.G. Graphic Representations of the Periodic System During One Hundred Years. Tuscaloosa: University of Alabama Press, 1974.
15. Scerri E. Periodic Table: its story and its significance. New York: Oxford University Press, 2019.
16. Haenzel G. Die Polygonfläche und das Periodische System der Elemente // Zeitschrift für Physik. 1943. V. 120. P. 283–300.
17. Finke W. Bemerkungen zu G. Haenzel „Die Polygonfläche und das Periodische System der Elemente.“ // Zeitschrift für Physik. 1943. V. 121. P. 586–587.
18. Фет А.И. Группа симметрии химических элементов. Новосибирск: Наука, 2010.
19. Варламов В.В. Теоретико-групповое описание периодической системы элементов // Математические структуры и моделирование. 2018. № 2 (46). С. 5–23.
20. Варламов В.В. Теоретико-групповое описание периодической системы элементов. II: Таблица Сиборга // Математические структуры и моделирование. 2019. № 1 (49). С. 5–21.
21. Варламов В.В. Теоретико-групповое описание периодической системы элементов. III: 10-периодическое расширение // Математические структуры и моделирование. 2019. № 3 (51). С. 5–20.
22. Ostrovsky V.N. Group theory and periodic system of elements // AIP Conference Proceedings. 1996. V. 365. P. 191–216.
23. Варламов В.В. О системе аксиом нелокальной квантовой теории // Математические структуры и моделирование. 2017. № 4 (44). С. 5–25.
24. Зоммерфельд А. Строение атома и спектры. М.: ГИТТЛ, 1956.

**GROUP THEORETICAL DESCRIPTION OF PERIODIC SYSTEM OF ELEMENTS:
VIII. THE STRUCTURE OF ENERGY LEVELS**

V.V. Varlamov

Dr.Sc. (Phys.-Math.), Professor, e-mail: varlamov@sibsiu.ru

Siberian State Industrial University, Novokuznetsk, Russia

Abstract. The structure of the energy levels of a periodic system is studied, the states of which (chemical elements) are represented as nodes of the weight diagram of the group algebra of the rotation group of an eight-dimensional pseudo-Euclidean space of the neutral signature.

Keywords: periodic table, Madelung rule, weight diagram, Laplace–Runge–Lenz vector, antimatter.

Дата поступления в редакцию: 24.12.2025

ЛОГИЧЕСКОЕ ДОКАЗАТЕЛЬСТВО РАЗРЕШИМОСТИ ПАРАДОКСА ДЕДУШКИ В РАМКАХ МОДАЛЬНОЙ ПРЕДИКАТНОЙ ЛОГИКИ ГЁДЕЛЯ S5 ВТОРОГО ПОРЯДКА

А.К. Гуц

д.ф.-м.н., профессор, e-mail: aguts@mail.ru

Сочинский государственный университет, Сочи, Россия

Аннотация. В данной статье рассматривается способ разрешения известного парадокса дедушки в рамках модальной предикатной логики S5 второго порядка, построенной Куртом Гёделем.

Ключевые слова: машина времени, модальная предикатная логика второго порядка, парадокс дедушки.

В статьях [1, 2] было приведено онтологическое доказательство (заключение, идущие от понятия к бытию) реальности, т. е. существования, или бытия Прошлого. Для этого мы использовали схему онтологического доказательства существования Бога, предложенного Гёделем в системе модальной логики S5 второго порядка [3].

Парадокс дедушки состоит в том, что если внук окажется в Прошлом до того, как дедушка встретил бабушку и убьёт его, то вопрос в том, что его, внука, не должно существовать, а значит, как разрешение возникающего противоречия состоит в том, что внук, попав в прошлое не может убить дедушку. Рассматривая парадокс дедушки, мы видим, что противоречие возникает только тогда, когда внук убивает дедушку. Часто вводятся постулаты, сводящиеся к заявлениям, что Мир устроен так, что подобное убийство невозможно. Однако такие постулаты требуют обоснования. Поэтому такое логическое доказательство мы ищем в рамках модальной логики. Считаем, что Прошлое обладает свойством завершённости во всем, т. е. всё, что прошло, то завершено, свершилось. Прошлое есть сущее, все атрибуты которого обладают завершёностью. С точки зрения абсолютной теории пространства-времени – это очевидный факт, изменений нет, они иллюзорны. Существование как реальность Прошлого в рамках модальной логики с учётом семантики Крипке есть высказывание

$$\Box \exists x P(x),$$

где $\Box$ – знак модальности «необходимо / обязательно», $P(x)$ – предикат, говорящий, что «объект x есть прошлое»; нам понадобится ещё другая модальность – $\Diamond$ «возможное». Введём предикат $Z(A)$ – атрибут (свойство) A «завершённое, свершившееся».

В [1, 2] доказана

Теорема 1. $\Box \exists x P(x)$, т. е. существование, бытие (реальность) Прошлого, необходимо / обязательно.

Это интерпретируем следующим образом: Прошлое, в котором живёт молодой дедушка до знакомства с бабушкой – атрибут A – это суть реальность, завершённая, свершившаяся, и, следовательно, неизменяемая. Поэтому, внук может появиться в прошлом молодого дедушки, но не сможет его убить.

Рассматривая парадокс дедушки, мы видим, что противоречие возникает только тогда, когда внук убивает дедушку. Часто вводятся постулаты, сводящиеся к заявлениям, что мир устроен так, что подобное убийство невозможно. Мы же концентрируем внимание на том, чтобы в рамках модальной логики доказать, что дедушка остаётся живым в любых обстоятельствах, на детали которых не следует обращать внимание. Иначе говоря, мы пишем формулу, которая утверждает, что внук ни в каком случае не остаётся без дедушки. Ведь если дедушка убит, то у убийцы-внука не может быть дедушки, поскольку он бы не родился. Иначе говоря, условие $\Box$ «не только верно сейчас, но условие $\Box$ будет верно при всех возможных развитиях ситуации» [4].

Пусть $V(x, y)$ — предикат, интерпретируем как « x – внук для y ». Тогда формулу [5]

$$\varphi = \forall x \exists y (V(x, y) \rightarrow \neg \Diamond \neg V(x, y)), \quad (1)$$

мы читаем как «у любого x есть дедушка y , и, следовательно, невозможно, чтобы y не был дедушкой x ».

Описание физической ситуации, физического процесса в использовании модальной логики – это написание последовательности формул на языке используемой логики, не допускающие их прочтения как истинных высказываний, которым следует следовать. Для этого нужно семантическое, т.е. осмысленное (означенное) их прочтение, наполняющее их конкретным и понятным содержанием. Иначе говоря, нужно указывать, как продемонстрировал Крипке, «возможные миры», в которых осмысление и происходит и когда можно говорить об истинности ранее написанных высказываний. Особое место здесь занимает означивание связок $\Box A$ (т.е. « A необходимо») и $\Diamond A$ (т.е. « A возможно»), когда говорят о достижимости нужного и желанного «мира». И важнейшим моментом является здесь указание на то, что понимается под «возможным миром».

О «возможных мирах» много говорят в аналитической философии. Так авторитетный аналитический философ Дэвид Льюис буквально понимал под ними альтернативные физические вселенные. Сам Крипке не склонялся к такой интерпретации. Логик ... из МГУ предпочитает говорить о различных состояниях одной (нашей) физической реальности, в которых мы оказываемся и в которых приходится осмысливать логическое высказывание.

Нам кажется, что в нашем случае, т.е. при рассмотрении пространства-времени в общей теории относительности с временными петлями (машины времени, как называл открывший их Курт Гёдель), «возможный мир» – это событие в пространстве-времени, или точка в лоренцевом многообразии, с помощью которого геометрически представлено пространство-время. Такой подход разрабатывает французский логик Халими [6], в какой-то мере, идущий по пути топологической интерпретации Тарского.

Ещё один важный вопрос – это в какой мере логические, т.е. сугубо абстрактные

мысленные построения соответствуют тому, что мы (физики, астрономы) наблюдаем в окружающем нас *реальном физическом Мире*? В нашем случае: можно ли принимать как истину то, что устанавливается логической формулой в конкретном логическом исчислении? Думается можно, поскольку мы используем дифференциальное исчисление – дифференциальные уравнения, и принимаем то, что они предсказывают, хотя они покоятся на весьма абстрактном понятии вещественного или комплексного числа, например как класс фундаментальных последовательностей или через дедекиндовы сечения. Число π мы принимаем, но выписать его не можем и не сможем никогда.

Литература

1. Гуц А.К. Реальность прошлого: эмпирические и онтологические доказательства // Математические структуры и моделирование. 2022. № 4 (64). С. 29–43.
2. Гуц А.К. Реальность прошлого и межвременных переходов в общей теории относительности // Основания фундаментальной физики и математики: материалы VI Российской конференции (ОФФМ–2022) / под ред. Ю.С. Владимирова, В.А. Панчелюги. Москва: РУДН, 2022. С. 168–173.
3. Пушкарский А.Г. Курт Гедель и его онтологическое доказательство // РАЦИО.ru. 2014. № 13. С. 153–172.
4. Артемов С.Н. Интуиционистская логика с точки зрения классической // Глобус. Общественно-математический семинар. М.: МЦНМО, 2004. С. 127–141.
5. Гуц А.К. Скачки во времени, предикатная модальная логика и разрешение парадокса дедушки // Математические структуры и моделирование. 2025. № 4 (76). С. 52–56.
6. Halimi B. Geometric Modal Logic // Notre Dame Journal of Formal Logic. 2023. V. 64/3. P. 377–406.

LOGICAL PROOF OF THE SOLVABILITY OF THE GRANDFATHER PARADOX IN THE FRAMEWORK OF GODEL'S MODAL PREDICATE LOGIC S5 OF THE SECOND ORDER

A.K. Guts

Professor, Dr.Sc. (Phys.-Math.), e-mail: aguts@mail.ru

Sochi State University, Sochi, Russia

Abstract. In this article the possibility of solving the well-known grandfather paradox within the framework of modal predicate logic is considered. Formulas are given that indicate the impossibility of generating a situation in which a contradiction is realized.

Keywords: time jumps, time machine, modal predicate logic, grandfather paradox.

Дата поступления в редакцию: 05.01.2026

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ СОСТОЯНИЙ В СПИНОВЫХ СИСТЕМАХ С НЕПРЕРЫВНОЙ СИММЕТРИЕЙ: АСПЕКТЫ ПРИМЕНЕНИЯ АЛГОРИТМА ВАНГА–ЛАНДАУ

Д.А. Друзьев

аспирант, e-mail: druzjevda@stud.omsu.ru

О.А. Терентьева

канд. техн. наук, доцент, e-mail: anatole4ka@yandex.ru

П.В. Прудников

доктор физ.-мат. наук, профессор, e-mail: prudnikovpv@omsu.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе показано, что методы генерации случайных состояний, дающие неравномерно распределённые точки на сфере, приводят к систематическим искажениям при моделировании методами Монте–Карло. На примере алгоритма Ванга–Ландау, предназначенного для прямого вычисления плотности состояний и микроканонической энтропии, продемонстрировано, что некорректный выбор метода генерации спиновых конфигураций вызывает смещение в оценке энтропии, что приводит к ошибкам в расчёте термодинамических характеристик. Сравнение с корректными методами генерации подтверждает критическую важность равномерного покрытия конфигурационного пространства при энтропийном моделировании.

Ключевые слова: Методы Монте–Карло, спиновые системы, случайные блуждания, модель Гейзенберга, алгоритм Метрополиса, алгоритм Ванга–Ландау.

Введение

Моделирование спиновых систем с непрерывной симметрией, таких как трёхмерная модель Гейзенберга, уже долгое время представляет значительный интерес в статистической физике и материаловедении. Модель позволяет эффективно описывать фазовые превращения в реальных магнетиках, учитывая непрерывное распределение спинов в пространстве и их взаимодействие между собой [1–3]. В классической модели Гейзенберга каждый спин представляется единичным трёхмерным вектором, лежащим на поверхности сферы S^2 :

$$\mathbf{S}_i = (S_i^x, S_i^y, S_i^z) \in \mathbb{R}^3, \quad \|\mathbf{S}_i\| = 1.$$

Гамильтониан классической изотропной модели без внешнего поля имеет вид:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j,$$

где S_i – спин в i -той ячейке трёхмерной решётки, а $\langle i, j \rangle$ означает, что сумма идёт по парам ближайших соседей, $J > 0$ соответствует ферромагнетизму. При критической температуре $T_c/Jk_B = 1.44292(8)$ [4] система испытывает фазовый переход второго рода ферромагнетик-парамагнетик.

1. Методы Монте–Карло

Методы Монте–Карло объединяет группу методов статистического моделирования, каждый из которых базируется на замене стохастических переменных сгенерированными случайными значениями и последующем усреднении результатов многих испытаний. Расчёт характеристик магнитных систем возможен на основе реализации различных алгоритмов.

Алгоритм Метрополиса [5] образует марковский процесс, на основе равновесного распределения Больцмана $P(E) \propto e^{-E/k_B T}$. Расчёт термодинамических величин происходит генерацией канонического распределения для фиксированной температуры T путём принятия/отказа нового случайного состояния B из состояния A с вероятностью:

$$P_{\text{accept}} = \min \left(1, \frac{P(E_B)}{P(E_A)} \right) = \min \left(1, e^{-(E_B - E_A)/k_B T} \right).$$

После достижения равновесия проводится наблюдение за интересующими величинами в течении τ_{obs} , затем усреднение:

$$\langle f(T) \rangle = \frac{\sum_{t=0}^{\tau_{obs}} f_t}{\tau_{obs}},$$

Подход, реализованный в **методе Ванга–Ландау** [6] позволяет напрямую рассчитать плотность состояний системы $g(E)$ – количество состояний с заданной энергией E . В процессе симуляции алгоритм адаптивно оценивает $g(E)$, блуждая по энергетическому спектру и уменьшая модификационный фактор. Вероятность принятия перехода не использует температуру напрямую, а задаётся как:

$$P_{\text{accept}} = \min \left(1, \frac{g(E_{\text{old}})}{g(E_{\text{new}})} \right).$$

После завершения работы алгоритма среднее значение термодинамической величины можно рассчитать проводя перевзвешивание по всевозможным уровням энергии:

$$\langle f(T) \rangle = \frac{\sum_E f(E) g(E) e^{-E/k_B T}}{\sum_E g(E) e^{-E/k_B T}},$$

Генерация случайного спина для пробного переворота это базовая операция, которая происходит на каждом элементарном шаге численного моделирования методами Монте–Карло и занимает основную часть вычислительных ресурсов, поэтому выбор метода генерации является важной частью исследования. В работе рассматриваются три метода генерации случайных спинов на единичной сфере, приведённые в таблице 1.

Выбор методов не случаен – **сферические координаты** являются очевидным представлением трёхмерного вектора единичной длины с помощью двух углов, но ввиду становления при переходе элемента площади ds пропорциональным $\sin \theta$:

$$ds = \sin \theta d\theta d\phi,$$

тогда плотность вероятности тоже должна быть пропорциональна $\sin \theta$, если же θ равномерна, то плотность постоянна: $p(\theta) = \frac{1}{\pi}$. В следствии чего точки будут накапливаться вблизи полюсов, где $\sin \theta \approx 0$ (рис. 1a).

Метод **Marsaglia** [7] был революционным в своё время из-за простоты и эффективности. Он основан на отображении единичного диска из $\mathbb{R}^2$ на сферу S^2 с сохранением равномерности и эквивалентен проекции из нормального распределения, но вычислительно эффективнее и является более предпочтительным среди других методов, ввиду генерации равномерно распределённых точек на сфере (рис. 1с) без использования медленных тригонометрических функций.

При **нормализации точек, равномерно распределённых внутри куба**, на сфере возникает неравномерность, так как точки скапливаются в направлениях главных диагоналей куба. Это связано с тем, что объём куба в этих направлениях больше (максимальное расстояние от центра до границы куба равно $\sqrt{3}$ против 1 вдоль осей), что приводит к большей плотности проецируемых точек после нормализации (рис. 1b).

Таблица 1. Способы генерации случайного спина. Где $\mathcal{U}[a, b]$ означает генерацию случайного равномерно распределённого вещественного числа в $[a, b]$.

Сферические координаты	Marsaglia	Нормализация точек в кубе
$\theta \sim \mathcal{U}[0, \pi], \phi \sim \mathcal{U}[0, 2\pi];$ $S_x = \sin \theta \cos \phi,$ $S_y = \sin \theta \sin \phi,$ $S_z = \cos \theta;$	$u_1, u_2 \sim \mathcal{U}[-1, 1];$ если $s = u_1^2 + u_2^2 \leq 1$: $S_x = 2u_1\sqrt{1-s},$ $S_y = 2u_2\sqrt{1-s},$ $S_z = 1-2s;$	$x, y, z \sim \mathcal{U}[-1, 1];$ $r = \sqrt{x^2 + y^2 + z^2}$ $S_x = x/r,$ $S_y = y/r,$ $S_z = z/r.$

2. Результаты

Для анализа влияния метода генерации случайных спинов в методах Монте–Карло был проведён численный расчёт теплоёмкости классической трёхмерной изотропной модели Гейзенберга:

$$C(T) = \frac{\langle E^2(T) \rangle - \langle E(T) \rangle^2}{k_B T^2};$$

В качестве оптимальных линейных размеров системы был выбран $L = 8$ и периодические граничные условия, для минимизации эффектов связанных с влиянием конечного размера решётки. Во время работы алгоритма Метрополиса выделялось 10^6 шагов Монте–Карло на спин (MCS/s) для достижения равновесия системы и

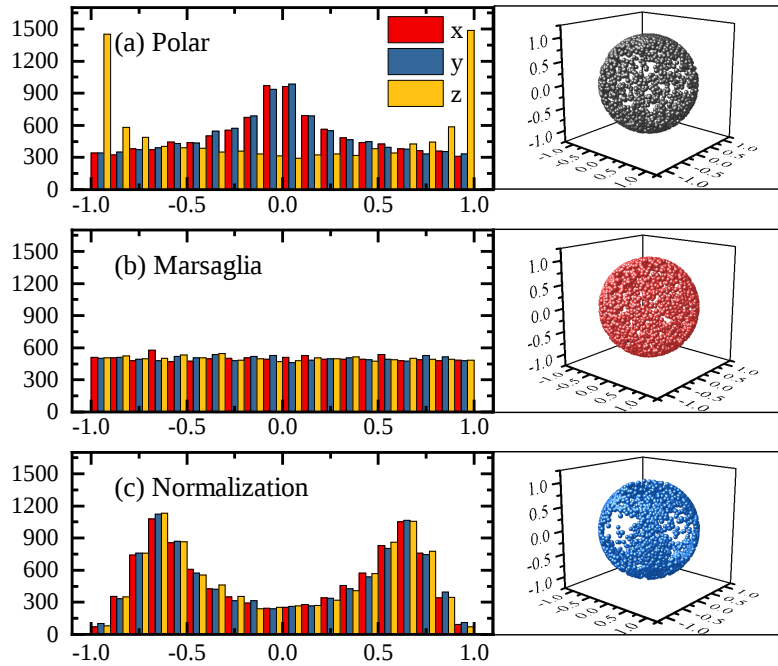


Рис. 1. Гистограммы распределения координат при генерации 10^5 точек на единичной сфере для различных методов генерации: (a) Сферические координаты, (b) Marsaglia, (c) Нормализация точек в кубе

10^3 шагов накопления данных для усреднения внутри одной статистической конфигурации, так же было проведено усреднение по 10^3 статистическим конфигурациям с различными инициализирующими значениями генератора случайных чисел. Для эффективного использования метода Ванга–Ландау была использована модификация $1/t$ [8], энергетический спектр был обрезан до $E_{min}/L^3 = -2.8J$ и разделён на отрезки размером $dE = 0.1J$, проведено 20 итераций алгоритма до достижения фактора сходимости $f_{finish} \approx 10^{-7}$. Данные были дополнительно усреднены по 20 независимым конфигурациям.

Результаты демонстрируют значительное расхождение в плотности состояний $g(E)$, полученной различными методами генерации, при приближении к низкотемпературной области (рис. 2a). В то же время, при использовании алгоритма Метрополиса различия между методами находятся в пределах статистической погрешности, которая уменьшается с ростом числа усредняемых конфигураций (рис. 2b). Особенно выделяются в алгоритме Ванга–Ландау неравномерные методы генерации, приводящие к смещению положения пиков теплоёмкости относительно результатов, полученных алгоритмом Метрополиса, тогда как метод Marsaglia, обеспечивающий равномерное распределение на сфере, демонстрирует хорошую сходимость (рис. 2c).

Это связано с фундаментальным различием в работе алгоритмов Метрополиса и Ванга–Ландау. Алгоритм Метрополиса устойчив к неравномерной генерации благодаря механизму детального баланса, который корректирует распределение через вероятность принятия/отклонения шага. Напротив, алгоритм Ванга–Ландау напря-

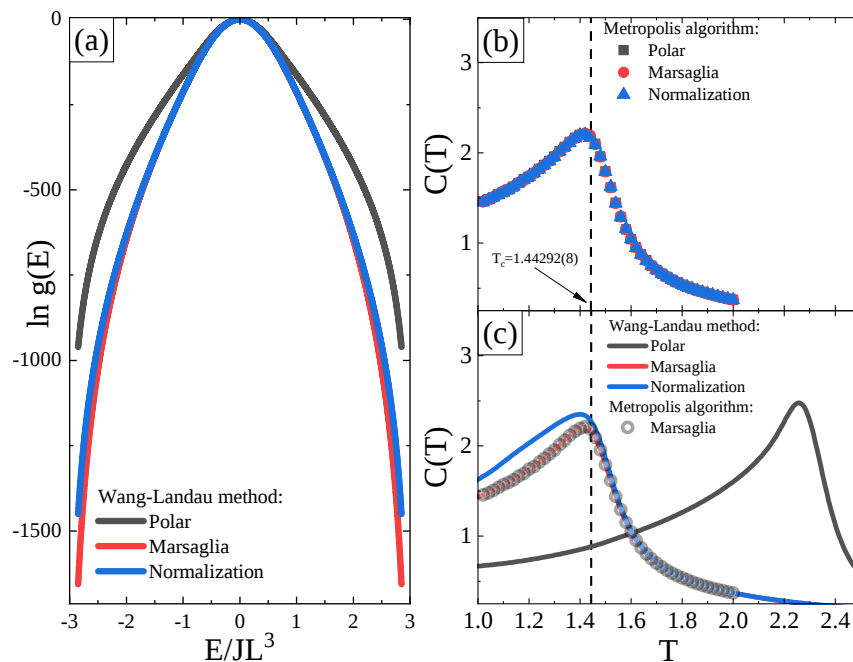


Рис. 2. Результаты работы методов Монте-Карло для различных методов генерации случайных спинов: (a) Плотности состояний, рассчитанные алгоритмом Ванга–Ландау, (b) Теплоёмкости $C(T)$, полученные алгоритмом Метрополиса, (c) Теплоёмкости $C(T)$, полученные алгоритмом Ванга–Ландау в сравнении с алгоритмом Метрополиса

мую рассчитывает $g(E)$, и его корректная работа требует равновероятного посещения всех микросостояний с фиксированной энергией. Вследствие нарушения этого условия методы с неравномерным распределением генерации продемонстрировали систематические ошибки в расчёте плотности состояний и описании пика теплоёмкости $C(T)$ вблизи критической точки модели. Такие же свойства можно ожидать для других алгоритмов, напрямую вычисляющих $g(E)$, таких как моделирование мультиканонического ансамбля [9] или методы гистограмм [10, 11].

Заключение

В данной работе были рассмотрены три различных способа генерации случайного спина на поверхности трёхмерной сферы в контексте моделирования спиновых систем методами Монте-Карло.

Благодаря механизму детального баланса, который гарантирует достижение распределения Больцмана вне зависимости от начальных условий, алгоритм Метрополиса устойчив к неравномерному посещению фазового пространства. Это свойство следует из фундаментальных принципов эргодичности и перемешивания в статистической механике [12]. Нарушение этих принципов может указывать на неприменимость методов к исследуемой модели [13].

Алгоритм Ванга–Ландау напрямую рассчитывает плотность состояний системы $g(E)$, и его применение требует равновероятного посещения всех микросостояний с фиксированной энергией. Как было показано, нарушение этого условия приводит

к систематическим ошибкам в оценке $g(E)$ и описании пика теплоёмкости вблизи критической температуры. Для таких алгоритмов метод Marsaglia [7] обеспечивает равномерное распределение точек на сфере и показывает свою высокую эффективность и надёжность. Этот способ также широко используется и в других методах Монте–Карло, включая генерацию случайных ориентаций и направлений, а также построение матриц вращения [14].

Выявление и анализ таких критически важных аспектов алгоритмов моделирования необходимо для глубокого понимания физических процессов в спиновых системах и разработки корректных теоретических моделей для описания фазовых переходов.

Литература

1. Janke W. Monte Carlo Simulations of Spin Systems // Computational Physics: Selected Methods, Simple Exercises, Serious Applications / ed. by K. H. Hoffmann, M. Schreiber. Berlin; Heidelberg: Springer, 1996. P. 10–43.
2. Druzev D.A., Chubarova A.A., Prudnikov P.V. Phase Transitions in Heisenberg Magnets Induced by Uniaxial Anisotropy: Wang–Landau and Machine Learning Simulations // JETP Letters. 2025. V. 122. P. 156–161.
3. Iakovlev I.A., Sotnikov O.M., Mazurenko V.V. Supervised learning approach for recognizing magnetic skyrmion phases // Physical Review B. 2018. V. 98, No. 17. P. 174411.
4. Chen K., Ferrenberg A.M., Landau D.P. Static critical behavior of three-dimensional classical Heisenberg models: A high-resolution Monte Carlo study // Physical Review B. 1993. V. 48, No. 5. P. 3249.
5. Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E. Equation of State Calculations by Fast Computing Machines // The Journal of Chemical Physics. 1953. V. 21, No. 6. P. 1087–1092.
6. Wang F., Landau D.P. Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States // Physical Review Letters. 2001. V. 86, No. 10. P. 2050–2053.
7. Marsaglia G. Choosing a Point from the Surface of a Sphere // The Annals of Mathematical Statistics. 1972. V. 43, No. 2. P. 645–646.
8. Belardinelli R.E., Pereyra V.D. Fast algorithm to calculate density of states // Physical Review E. 2007. V. 75, No. 4. P. 046701.
9. Berg B.A., Neuhaus T. Multicanonical ensemble: A new approach to simulate first-order phase transitions // Physical Review Letters. 1992. V. 68, No. 1. P. 9–12.
10. Kastner M., Promberger M., Muñoz J.D. Broad histogram method: Extension and efficiency test // Physical Review E. 2000. V. 62, No. 5. P. 7422–7429.
11. Ferrenberg A.M., Swendsen R.H. New Monte Carlo technique for studying phase transitions // Physical Review Letters. 1988. V. 61, No. 23. P. 2635–2638.
12. Brazhkin V.V. Why does statistical mechanics 'work' in condensed matter? // Phys. Usp. 2021. V. 64, No. 64. P. 1049–1057.
13. Стронгин В.С., Шевченко Ю.А., Нефедев К.В. Нарушение эргодичности фрустрированных спиновых систем // Дальневосточный математический журнал. 2026. Вып. 26, № 2. (в печати).

14. Schnabel S., Janke W. A simple algorithm for uniform sampling on the surface of a hypersphere. arXiv preprint arXiv:2204.14004. 2022. URL: <https://arxiv.org/abs/2204.14004> (дата обращения: 30.01.2026).

RANDOM STATE GENERATION IN CONTINUOUS SPIN SYSTEMS: ASPECTS OF THE WANG–LANDAU ALGORITHM APPLICATION

D.A. Druzev

Ph.D. Student, e-mail: druzjevda@stud.omsu.ru

O.A. Terentyeva

Ph.D. (Techn.), Associate Professor, e-mail: anatole4ka@yandex.ru

P.V. Prudnikov

Dr.Sc. (Phys.-Math.), Professor, e-mail: prudnikovpv@omsu.ru

Dostoevsky Omsk State University, Omsk, Russia

Abstract. This paper demonstrates that random state generation methods that produce nonuniformly distributed points on a sphere lead to systematic errors in Monte Carlo simulations. Using the Wang–Landau algorithm, designed for directly calculating the density of states and microcanonical entropy, it is demonstrated that an incorrect choice of spin configuration generation method leads to a bias in the entropy estimate, leading to errors in the calculation of thermodynamic characteristics. A comparison with correct generation methods confirms the critical importance of uniformly covering the configuration space in entropy methods.

Keywords: Monte–Carlo methods, spin systems, random walks, Heisenberg model, Metropolis algorithm, Wang–Landau method .

Дата поступления в редакцию: 30.01.2026

К ДОКАЗАТЕЛЬСТВУ ЛЕММЫ О ЛОКАЛЬНОМ ВРАЩЕНИИ

В.И. Субботин

к.ф.-м.н., доцент, e-mail: geometry@mail.ru

Южно-Российский государственный политехнический университет им. М.И. Платова,
Новочеркасск, Россия

Аннотация. В работе приведено обновлённое доказательство так называемой леммы о локальном вращении. Так автор называет доказанное им ранее утверждение о выпуклых многогранниках в E^3 с условием локальной симметричности граней. При этом грань F многогранника называется локально симметричной, если ось вращения грани F , перпендикулярная ей, является осью вращения гранной звезды $\text{Star}(F)$. Звезда $\text{Star}(F)$ – это совокупность граней многогранника, каждая из которых имеет общее ребро с гранью F . Лемма о локальном вращении утверждает, что каждая локально симметричная грань является глобально симметричной, то есть ось вращения грани F будет осью вращения многогранника в целом.

Ключевые слова: звезда грани, локальное вращение, условие симметричности

Введение

Изучение классов многогранников в E^3 с различными условиями симметрии является важной задачей классической и современной геометрии, (см. [1] – [3], [6] гл. 6). Первыми известными классами многогранников с условиями симметричности являются пять правильных многогранники (платоновых) и тринадцать полуправильных (архимедовых). Выпуклые правильные многогранники Джонсона (см. [4], [5]) можно тоже считать многогранниками с условиями симметричности. Заметим, что для удобства, полезно различать слабые и сильные условия симметричности. Например, упомянутые классы правильных и полуправильных многогранников следует считать классами с *сильными* условиями симметричности, то есть в этих классах только конечное число представителей. Класс многогранников Джонсона является классом со *слабыми* условиями симметричности, так как условие правильности здесь приводит, в частности, к бесконечным сериям призм и антипризм.

Ранее автором был полностью описан класс со слабым условием симметричности так называемых выпуклых FS – многогранников, ([7]). FS – многогранником был назван такой, у которого каждая грань имеет центр вращения, через который проходит нетривиальная ось вращения L всего многогранника. Если не требовать, чтобы каждая ось L была осью вращения всего многогранника, а только лишь осью вращения гранной звезды соответствующей грани, то мы получим тот же самый

класс FS . Для доказательства эквивалентности локального и глобального условия симметричности для класса FS нужна была *лемма о локальном вращении*, (см., например, [8]).

В настоящей работе доказательство леммы о локальном вращении будет уточнено и сделано более наглядным.

1. Определения

Определение 1. Звездой $Star(F)$ грани F многогранника называется множество всех граней, имеющих общее ребро с гранью F .

Определение 2. Грань F многогранника назовём локально симметричной, если у неё есть нетривиальная ось вращения, перпендикулярная F и являющаяся осью вращения звезды $Star(F)$.

Определение 3. Грань F многогранника назовём глобально симметричной, если у неё есть нетривиальная ось вращения, перпендикулярная F и являющаяся осью вращения всего многогранника.

Определение 4. Ось вращения, порядок которой больше 1, называется нетривиальной.

Замечание 1. Порядок оси вращения грани многогранника может быть меньше порядка её оси вращения как фигуры, отделённой от многогранника.

Определение 5. Поясом P граней называется связное множество граней $F_1, F_2, \dots, F_n$ такое, что любые две грани F_i, F_{i+1} и также F_1, F_n имеют только одно общее ребро, а любые две другие грани множества P не имеют общих ребер.

Определение 6. Множество поясов P_i на поверхности многогранника называется *связным*, если от одного пояса к другому можно перейти по граням поясов.

Определение 7. Связное множество поясов далее называется *цепью* граней.

2. Лемма о локальном вращении

Лемма 1. Если все грани замкнутого выпуклого многогранника в E^3 являются локально симметричными, то они являются также и глобально симметричными.

Доказательство. Рассмотрим многогранник M , каждая грань F_i которого обладает свойством *локальной симметрии*, то есть при некотором вращении, при котором грань F_i самосовмещается, будет самосовмещаться звезда $Star(F_i)$. Для доказательства достаточно показать, что при локальном самосовмещении грани F_i многогранник M самосовместится.

Построим цепь, состоящую из двух граней. На рисунке показан пример цепи из правильных треугольников и квадратов. Пусть F_1 — некоторая грань многогранника M , Рис. 1. Пусть $Star(F_1)$ состоит из трёх граней G_1 . Пользуясь свойством

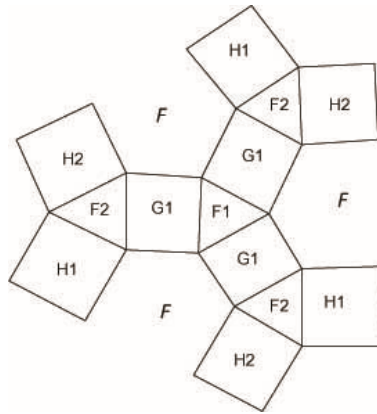


Рис. 1. К доказательству леммы

локальной симметрии граней G_1 , получим, что при вращении грани F_1 , при котором она самосовмещается, будут совмещаться грани F_2 . Аналогично, пользуясь свойством локальной симметрии граней F_2 , получим, что при локальной симметрии грани F_1 , будут совмещаться звёзды граней F_2 .

Продолжая этот процесс далее, получим замкнутую цепь C , в которой две различные грани. Заметим, что в частности, в C все грани могут быть равными между собой. При самосовмещении любой грани T цепи C вместе с соседними гранями с T в C вся цепь переходит в себя. В результате построения цепи C возможны два случая:

- 1) C составит всю поверхность многогранника M ;
- 2) останутся части поверхности многогранника M , не входящие в C , обозначенные на рисунке 1 как F .

В первом случае утверждение леммы очевидно. Рассмотрим второй случай. Так как при самосовмещении любой грани цепи C вся цепь C переходит в себя, то области F , очевидно, переходят в области F . Причём, граница каждой F состоит из некоторых рёбер и вершин граней, входящих в C . Здесь так же возможны всего два случая.

- 1) Граница области F состоит из рёбер только одной грани цепи C ;
- 2) граница области F состоит из рёбер двух возможных граней цепи C .

В первом случае рассмотрим некоторую грань $\Phi \notin C$, которая имеет общее ребро с некоторой гранью $G_1 \in C$. Очевидно, $\Phi \in F$ и $G_1 \in \text{Star}(\Phi)$.

Построим цепь D из двух граней G_1 и Φ . При вращении и самосовмещении Φ вся цепь D будет переходить в себя. Следовательно, границей грани Φ будет вся граница области F , которая будет состоять из рёбер граней, эквивалентных G_1 . Значит, область F совпадает с гранью Φ . При вращении и самосовмещении Φ не только цепь D , но и цепь C будет тоже переходить в себя. Действительно, все грани, эквивалентные G_1 входят также и в цепь C . При указанном вращении грани Φ , все оставшиеся грани, эквивалентные F_1 , будут также переходить в себя, то есть при самосовмещении Φ самосовместится не только цепь D , но и цепь C ; таким образом, самосовместится весь многогранник M .

Второй случай, когда граница области F состоит из рёбер двух возможных гра-

ней цепи C , рассматривается аналогично. В этом случае в $Star(\Phi)$ входят грани, эквивалентные $G1$ и $F1$. Построив две цепи, D и E , где E — цепь, построенная из двух граней, Φ и $F1$. При самосовмещении Φ каждая из цепей D и E будет переходить в себя. Так как при самосовмещении Φ вся область F переходит в себя, а граница Φ состоит из рёбер граней, эквивалентных $G1$ и $F1$, то как и в первом случае, F совпадает с Φ .

Таким образом, при любом локальном вращении любой грани многогранника M , все грани многогранника самосовмещаются, то есть самосовмещается сам многогранник M . ■

Из доказанной леммы получаем

Следствие 1. *В многограннике, в котором каждая грань является локально симметричной, количество различных граней может быть 1, 2 или 3.*

Литература

1. Cromwell P.R. Polyhedra. Cambridge: Cambridge University Press, 1999. 451 p.
2. Coxeter H.S.M. Regular Polytopes. Dover Publications, 1973. 321 p.
3. Grunbaum B. Regular polyhedra—old and new // Aequationes mathematicae. 1977. V. 16, No. 1–2. P. 1–20.
4. Jonson N.W. Convex polyhedra with regular faces // Canadian Journal of Mathematics. 1966. V. 18, No. 1. P. 169–200.
5. Залгаллер В.А. Выпуклые многогранники с правильными гранями // Записки научных семинаров ЛОМИ. 1967. Т. 2. С. 1–220.
6. Изометрические полиэдральные подграфы в гиперкубах и кубических решетках / М. Деза [и др.]. М.: МЦНМО, 2008. 192 с.
7. Субботин В. И. Об одном классе сильно симметричных многогранников // Чебышевский сб. 2016. Т. 17, № 4. С. 132–140.
8. Субботин В.И. О многогранниках, сильно симметричных относительно вращения // Чебышевский сборник. 2006. Т. VII, № 2 (18). С. 168–171.

TO THE PROOF OF THE LOCAL ROTATION LEMMA

V.I. Subbotin

Ph.D. (Phys.-Math.), Associate Professor, e-mail: geometry@mail.ru

Platov South Russian State polytechnic university, Novocherkassk, Russia

Abstract. This paper presents an updated proof of the so-called local rotation lemma. This is the author's name for a previously proven assertion about convex polyhedra in E^3 with local face symmetry. A face F of a polyhedron is said to be locally symmetric if the rotation axis of F , perpendicular to it, is the rotation axis of the face star $Star(F)$. A star $Star(F)$ is the set of faces of the polyhedron, each of which shares an edge with F . The local rotation lemma asserts that every locally symmetric face is globally symmetric, meaning that the rotation axis of F is the rotation axis of the polyhedron as a whole.

Keywords: star of the face, local rotation, symmetry condition.

Дата поступления в редакцию: 23.11.2025

МИРОВЫЕ ЛИНИИ СВЕТОВЫХ ЧАСТИЦ НА «КОНУСАХ ПРОШЛОГО» ДЛЯ НЕКОТОРЫХ НЕРЕЛЯТИВИСТСКИХ ПОРЯДКОВ В R^3

Н.Л. Шаламова

к.ф.-м.н., доцент, e-mail: parsifal8@inbox.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В данной статье изучаются мировые линии световых частиц «конусов прошлого» для некоторых порядков, инвариантных относительно некоммутативных разрешаемых групп Ли в их аффинном представлении. Для этих порядков строятся «конусы прошлого» и в некоторых случаях интегрируются уравнения геодезических.

Ключевые слова: частичный порядок, «конусы прошлого», геодезические линии, несвязные порядки.

Данная работа относится к направлению в геометрии, начало которой положил академик А.Д. Александров. Это направление принято называть хроногеометрией [1–5]. Задачами хроногеометрии занимались такие математики, как А.К. Гуц, А.В. Шайденко, А.В. Левичев и другие.

А.К. Гуц написал монографию по данной тематике, в которой изложена абсолютно полная информация по указанному геометрическому направлению [6, 7].

Введём следующие обозначения и определения. Будем говорить что в n -мерном арифметическом пространстве R^n ($n \geq 2$) задан частичный порядок $\mathcal{P}$, если задано семейство подмножеств $\mathcal{P} = \{P_x : x \in R^n\}$ пространства R^n такое, что выполнены следующие условия: 1) $x \in P_x$; 2) если $y \in P_x$, то $P_y \subset P_x$; 3) если $x \neq y$, то $P_x \not\subset P_y$ [6, 8]. Назовём порядок $\mathcal{P}$ связным, если $x \in \overline{P_x \setminus \{x\}}$, иначе порядок $\mathcal{P}$ будет несвязным.

Внешним конусом C_x множества P_x назовём замыкание объединения лучей l_{xy} , выходящих из точки x и проходящих через все $y \in P_x$, т. е. [6]

$$C_x = \overline{\bigcup_{y \in P_x} l_{xy}}.$$

Замечание 1. Предполагается, что на R^n задана естественная типология. Через $\overline{A}$ обозначаем замыкание множества A ; через ∂A – границу множества A .

Будем говорить, что внешний конус C_x – это конус с «острой вершиной», если C_x не содержит прямой.

«Конусом прошлого» для любого $P_x \in \mathcal{P}$ назовём следующие множество [6, 9]:

$$P_x^- = \{y \in R^n : x \in P_y\}.$$

Назовём гомеоморфизм $f : R^n \rightarrow R^n$ порядковым $\mathcal{P}$ -автоморфизмом, если $f(P_x) = P_{f(x)}$. Очевидно что если $\mathcal{P}$ -порядок $\mathcal{P}$ -инвариантен относительно группы G , то $G \subseteq \text{Aut}(\mathcal{P})$, где через $\text{Aut}(\mathcal{P})$ обозначим группу $\mathcal{P}$ -автоморфизмов.

Указанная выше разрешимая группа Ли G имеет вид:

$$\bar{x}_i = x_i e^{-\alpha_i t} + s_i,$$

где $(i = 1, \dots, n-1)$; $\bar{x}_n = x_n + t$; $\alpha_i \geq 0$; $\sum_{i=1}^n \alpha_i^2 \geq 0$ [6].

Далее в статье исследуются такие порядки $\mathcal{P}$, что множества P_x , $x \in R^n$ задаются аналитически, а не описательно, т. е. сначала задаём $P_e = \{(x_1, \dots, x_n) \in R^n : x_n \geq F(x_1, \dots, x_{n-1})\}$ для связного порядка; $P_e = \{(0, \dots, 0)\} \cup \{(x_1, \dots, x_n) \in R^n : x_n \geq F(x_1, \dots, x_{n-1})\}$ для несвязного порядка. Здесь и далее $e = (0, \dots, 0)$. Затем остальные множества P_x , $x \in R^n$, $x \neq e$ получаем преобразованиями из группы G . Здесь $F(x_1, \dots, x_{n-1})$ – некоторая дифференцируемая (за исключением, быть может, конечного числа точек) функция.

Приведём пояснительные примеры.

Пример 1.

$$P_e = \{(x_1, \dots, x_n) \in R^n : x_n \geq \sqrt[n]{x_1^2 + \dots + x_{n-1}^2}\};$$

$$P_z = \{(x_1, \dots, x_n) \in R^n : x_n - t \geq \sqrt{\sum_{i=1}^{n-1} ((x_i - s_i)e^{\alpha_i t})^2}\},$$

$$z = (s_1, s_2, \dots, s_{n-1}; t).$$

Пример 2. Рассмотрим несвязный порядок, где

$$P_e = \{(0, \dots, 0)\} \cup \{(x_1, \dots, x_n) \in R^n : x_n \geq 2 - e^{-\sum_{i=1}^{n-1} x_i^2}\};$$

$$P_z = \{(s_1, \dots, s_{n-1}; t)\} \cup \{(x_1, \dots, x_n) \in R^n : x_n - t \geq 2 - e^{-\sum_{i=1}^{n-1} ((x_i - s_i)e^{\alpha_i t})^2}\}.$$

Поскольку по определению $P_x^- = \{y \in R^n : x \in P_y\}$ и наш порядок инвариантен относительно группы G , то достаточно найти P_e^- , а остальные «конусы прошлого» P_x^- получить преобразованиями из группы G . Имеем

$$P_e = \{(x_1, \dots, x_n) \in R^n : x_n \geq F(x_1, \dots, x_{n-1})\}$$

для связного порядка; тогда

$$P_e^- = \{(s_1, \dots, s_{n-1}; t) \in R^n :$$

$$-t \geq F((-s_1 e^{\alpha_1 t}); (-s_2 e^{\alpha_2 t}); \dots; (-s_{n-1} e^{\alpha_{n-1} t})), t \leq 0\},$$

т. е. в функции $F(x_1, \dots, x_{n-1})$ координате x_i ($i = 1, \dots, n-1$) присвоено значение $(-s_i e^{\alpha_i t})$. Если перейти к привычным обозначениям, то

$$P_e^- = \{(x_1, \dots, x_n) \in R^n : x_n \leq 0; -x_n \geq F((-x_1 e^{\alpha_1 x_n}), \dots, (-x_{n-1} e^{\alpha_{n-1} x_n}))\}.$$

для связного порядка. Для несвязного порядка

$$P_e^- = \left\{ (0, \dots, 0) \right\} \cup \left\{ (x_1, \dots, x_n) \in R^n : \right. \\ \left. x_n \leq 0; -x_n \leq F((-x_1 e^{\alpha_1 x_n}), \dots, (-x_{n-1} e^{\alpha_{n-1} x_n})) \right\}.$$

Для примера 1.

$$P_e = \left\{ (x_1, \dots, x_n) \in R^n : x_n \geq \sqrt[4]{\sum_{i=1}^{n-1} x_i^2} \right\}.$$

Тогда

$$P_e^- = \left\{ (x_1, \dots, x_n) \in R^n : -x_n \geq \sqrt[4]{\sum_{i=1}^{n-1} x_i e^{\alpha_i x_n}} \right\}.$$

P_x^- , где $z = (z_1, \dots, z_n)$, имеет вид:

$$P_x^- = \left\{ (x_1, \dots, x_n) \in R^n : -x_n - z_n \geq \sqrt[4]{\sum_{i=1}^{n-1} (x_i - z_i)^2 e^{2\alpha_i(x_n - z_n)}}, z_n \neq x_n \right\}.$$

Отметим, что для получения всех выше изложенных результатов автор статьи использовала геометрические и теоретико-множественные конструкции. Полученные результаты можно оформить как теорему:

Теорема. Пусть $\mathcal{P} = \{P_x : x \in R^n; n \geq 2\}$ – частичный порядок в R^n , для которого выполняются следующие условия:

1. $\mathcal{P}$ инвариантен относительно преобразований из группы G .
2. Множество P_e ($e = (0, \dots, 0)$) заданно таким образом:
 - (а) для связного порядка $P_e = \{(x_1, \dots, x_n) \in R^n : x_n \geq F(x_1, \dots, x_{n-1})\}$;
 - (б) для несвязного порядка $P_e = \{(0, \dots, 0)\} \cup \{(x_1, \dots, x_n) \in R^n : x_n \geq F(x_1, \dots, x_{n-1})\}$, где $F(x_1, \dots, x_{n-1})$ – дифференцируемая, за исключением, быть может, конечного числа точек, функция.

Тогда для случая (а) «конус прошлого»

$$P_e^- = \left\{ (x_1, \dots, x_n) \in R^n : x_n \leq 0; x_n + F(z_1, \dots, z_{n-1}) \leq 0; \right. \\ \left. z_i = -x_i e^{\alpha_i x_n}, i = 1, \dots, n-1 \right\}.$$

Для несвязного порядка (б)

$$P_e^- = \left\{ (0, \dots, 0) \right\} \cup \left\{ (x_1, \dots, x_n) \in R^n : \right. \\ \left. x_n \leq 0; x_n + F(z_1, \dots, z_{n-1}) \leq 0; z_i = -x_i e^{\alpha_i x_n}; i = 1, \dots, n-1 \right\}.$$

«Конусы прошлого» для $x \neq e$ получается из P_x^- преобразованные из группы G .

Естественно предположить, что свет от одной точки до другой точки идёт по кратчайшей траектории. Поэтому геодезические линии, как кратчайшие на каком-то (возможно небольшом) промежутке, явно являются претендентами на мировые линии световых частиц [10, 11].

Рассмотрим случай где $n = 3$. Пусть $\partial P_x^- \setminus \{e\}$ является поверхностью вращения во круг оси x_3 , данную ось далее будем обозначать z ($x_1 = x$; $x_2 = y$). Если подходящей параметризацией метрический тензор поверхности $z = -F(-xe^{\alpha z}, -ye^{\alpha z})$ можно привести к виду $ds^2 = g_{11}(u)du^2 + g_{22}(v)dv^2$, где u, v – внутренние координаты, то система дифференциальных уравнений для нахождения геодезических имеет вид:

$$\begin{cases} 2g_{11}\frac{d^2u}{dt^2} + (g_{11})_u\left(\frac{du}{dt}\right)^2 - (g_{22})_u\left(\frac{dv}{dt}\right)^2 = 0, \\ g_{22}\frac{d^2v}{dt^2} + (g_{22})_u\frac{du}{dt}\frac{dv}{dt} = 0. \end{cases}$$

Частное решение этой системы $v = v_0 = \text{const}$ имеет вид $C_1t + C_2 = \int \sqrt{g_{11}}du$ – это, очевидно, меридианы поверхности вращения. Если $\frac{dv}{dt} \neq 0$, то получается решение

$$v + C = \pm \int \sqrt{\frac{g_{11}}{g_{22}(C_1g_{22} - 1)}} du.$$

В первом из приведённых примеров $P_e = \{(x, y, z) \in R^3 : z \geq \sqrt[4]{x^2 + y^2}\}$. «Конус прошлого» $\partial P_e^- \setminus \{e\}$ для группы $G_{III}V$ по классификации Бианки $\bar{x} = xe^{-t} + s$, $\bar{y} = ye^{-t} + l$; $\bar{z} = z + t$ имеет вид:

$$\partial P_e^- \setminus \{e\} = \{(x, y, z) : z < 0, z = e^{\frac{z}{2}} \sqrt[4]{x^2 + y^2}\}.$$

Введём параметризацию: $x = u^2e^{-u} \cos v$; $y = u^2e^{-u} \sin v$, $z = u$. Метрический тензор в этом случае имеет вид

$$ds^2 = e^{-2u}((e^{2u} + u^2(u - 2)^2)du^2 + u^4dv^2).$$

Тогда первое семейство геодезических – это всегда меридианы поверхности вращения $v = v_0$, т. е. $x = u^2e^{-u} \cos v_0$, $y = u^2e^{-u} \sin v_0$, $z = u$. Если $v \neq \text{const}$, то внутренние уравнения геодезических имеют вид

$$V + C = \pm \int \frac{e^u \sqrt{e^{2u} + u^2(2 - u)^2}}{u^2 \sqrt{c_1 u^2 - e^{2u}}} du.$$

Как будут выглядеть данные кривые на световом «конусе прошлого», сказать затруднительно.

Замечание 2. «Конусы прошлого» P_x^- , $x \in R^n$ также задают порядок на R^n . Однако внешние конусы C_x^- для P_x^- не будут конусами «с острой вершиной». Они будут содержать прямые даже в том случае, когда C_x , $x \in R^n$ были конусами с острой вершиной, т. е. прямых не содержали.

Литература

1. Alexandrov A.D. Contribution to chronogeometry // Canadian Journal of Mathematics. 1967. V. 19, No. 6. P. 1119–1128.
2. Александров А.Д. К основаниям геометрии пространства-времени // Доклады АН СССР. 1974. Т. 219, № 2. С. 265–267.
3. Александров А.Д. Отображения упорядоченных пространств // Труды Математического института АН СССР. М., 1972. Т. 128. С. 3–21.
4. Гуц А.К. Хроногеометрия профессора Ю.Ф. Борисова // Математические структуры и моделирование. 2025. № 2 (74). С. 5–11.
5. Borchers H.J., Hegerfeldt G.C. The structure of space–time transformations // Communications in Mathematical Physics. 1972. V. 28, No. 3. P. 259–266.
6. Гуц А.К. Хроногеометрия. Аксиоматическая теория относительности. Омск: Изд-во Ом. гос. ун-та, 2008. 340 с.
7. Гуц А.К. Теории пространства-времени // Пространство, время и фундаментальные взаимодействия. 2019. № 4. С. 23–47.
8. Шаламова Н.Л. Асимптотически линейчатый порядок // Математические структуры и моделирование. 2001. № 7. С. 56–60.
9. Гуц А.К. Реальность прошлого: эмпирические и онтологические доказательства // Математические структуры и моделирование. 2022. № 4 (64). С. 29–43.
10. Пименов Р.И. К основаниям теории дифференцируемого пространства-времени // Доклады АН СССР. 1975. Т. 222, № 1. С. 36–38.
11. Дубровин Б.А., Новиков С.П., Фоменко А.Т. Современная геометрия. Методы и приложения // Эдиториал УРСС. М., 1998. 336 с.

WORLDLINES OF LIGHT PARTICLES ON THE "PAST CONES" FOR CERTAIN NON-RELATIVISTIC ORDERS IN R^3

N.L. Shalamova

Ph.D. (Phys.-Math.), Associate Professor, e-mail: parsifal8@inbox.ru

Dostoevsky Omsk State University, Omsk, Russia

Abstract. In this paper, we study the worldlines of light particles on the "past cones" for certain orders invariant under noncommutative solvable Lie groups in their affine representation. For these orders, we construct the "past cones" and, in several cases, integrate the geodesic equations.

Keywords: Partial order, "Past cones", Geodesic lines, Disconnected orders.

Дата поступления в редакцию: 10.11.2025

ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ДЛЯ ОПТИМИЗАЦИОННОЙ ВЕРСИИ ЗАДАЧИ ГРАФ-ПОДГРАФ ИЗОМОРФИЗМА

Ю.В. Захарова^{1, 2}

к.ф.-м.н., старший научный сотрудник, e-mail: yzakharova@ofim.oscsbras.ru

В.П. Лобанов^{1, 2}

аспирант, e-mail: lobanov.vasiliy.pavlovich@gmail.com

Г.Р. Кравцов³

студент, e-mail: grk55rus@mail.ru

¹Омский филиал Института математики им. С.Л. Соболева СО РАН, Омск, Россия

²Новосибирский государственный университет

³Омский государственный технический университет

Аннотация. В статье предлагаются эволюционные алгоритмы для оптимизационной версии задачи граф-подграф изоморфизма, которая имеет приложения в компьютерных системах и в биоинформатике. Генетический алгоритм использует операторы, основанные на наследовании или мутации значений в позициях перестановок, используемых для кодировки решений. Эволюционная стратегия задействует адаптивный выбор операторов. Экспериментальные исследования показывают применимость разработанных алгоритмов.

Ключевые слова: эволюционная стратегия, генетический алгоритм, изоморфизм, эксперимент.

Введение

В статье рассматривается оптимизационный вариант задачи граф-подграф изоморфизма, где по заданным двум помеченным графам (малому и большому) требуется найти «максимально похожий» на малый граф подграф в большом графе [4, 6]. Рассматриваемая задача имеет практическое приложение в различных областях. Одной из которых является биоинформатика, где сопоставление сетей (Network Alignment) используется для выявления структурных сходств между биологическими связями. Это позволяет переносить свойства функций белков и генов от хорошо изученных видов к менее изученным. Такой подход используется при анализе взаимодействия белков (PPI), выравнивании геномных последовательностей и метаболических сетях [6]. Кроме того, взвешенный вариант задачи граф-подграф изоморфизма возникает в системах распределённых вычислений в задачах рационального использования ресурсов [4]. При распределении нагрузки требуется эффективно распределять операции между вычислительными узлами, учитывая такие параметры, как интенсивность обмена данными и загрузку процессоров. Ещё одна область применения – оптимизация работы многопроцессорных и серверных систем, где

требуется размещать операции с учётом структуры связей между компонентами [4]. Аналогичные задачи возникают в транспортных, биологических и социальных сетях.

В статье [6] авторы для решения оптимизационной версии задачи граф-подграф изоморфизма разработали гибридный меметический алгоритм, сочетающий генетический алгоритм и локальный поиск. Алгоритм переборного типа представлен в [8].

В настоящей работе предлагается адаптивная эволюционная стратегия для решения задачи граф-подграф изоморфизма. В алгоритме для адаптивного выбора операторов используется обучение с подкреплением, что позволяет находить решения хорошего качества при совместной работе операторов.

1. Постановка задачи

Даны два неориентированных графа $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$, где V_i – множество вершин, E_i – множество рёбер. Здесь $|V_2| \leq |V_1|$. Требуется найти такое сопоставление $\varphi : V_2 \rightarrow V_1$, что функция

$$f(\varphi, G_1, G_2) = \frac{1}{2} \sum_{u,v \in V_2} \tau_\varphi(u, v)$$

принимает максимальное значение, где

$$\tau_\varphi(u, v) = \begin{cases} 1, & \text{если } \{u, v\} \in E_2 \text{ и } \{\varphi(u), \varphi(v)\} \in E_1, \\ 0 & \text{иначе.} \end{cases}$$

Данная задача является NP -трудной, так как к ней полиномиально сводится задача об изоморфном подграфе (см., например, [6]).

Далее составим модель целочисленного линейного программирования (ЦЛП). Введём булевы переменные x_{ui} такие, что

$$x_{ui} = \begin{cases} 1, & \text{если } \varphi(u) = i, \\ 0, & \text{иначе,} \end{cases} \quad u \in V_2, i \in V_1.$$

Также введём булевы переменные w_{uvij} такие, что

$$w_{uvij} = \begin{cases} 1, & \text{если } \varphi(u) = i \text{ и } \varphi(v) = j, \\ 0, & \text{иначе,} \end{cases} \quad \{u, v\} \in E_2, \{i, j\} \in E_1.$$

Модель ЦЛП для задачи может быть записана следующим образом:

$$\sum_{\{u,v\} \in E_2} \sum_{\{i,j\} \in E_1} w_{uvij} \rightarrow \max, \quad (1)$$

$$\sum_{i \in V_1} x_{ui} = 1, \quad \forall u \in V_2, \quad (2)$$

$$\sum_{u \in V_2} x_{ui} \leq 1, \quad \forall i \in V_1, \quad (3)$$

$$w_{uvij} \leq x_{ui}, \quad \forall \{u,v\} \in E_2, \forall \{i,j\} \in E_1, \quad (4)$$

$$w_{uvij} \leq x_{vj}, \quad \forall \{u,v\} \in E_2, \forall \{i,j\} \in E_1, \quad (5)$$

$$w_{uvij} \geq x_{ui} + x_{vj} - 1, \quad \forall \{u,v\} \in E_2, \forall \{i,j\} \in E_1, \quad (6)$$

$$x_{ui} \in \{0, 1\}, \quad w_{uvij} \in \{0, 1\}, \quad \forall u, v \in V_2, \forall i, j \in V_1. \quad (7)$$

Целевая функция (1) максимизирует число рёбер графа G_2 , которые при сопоставлении φ сохраняются, то есть переходят в рёбра графа G_1 . Ограничение (2) гарантирует, что отображение φ определено для всех вершин G_2 : каждая вершина выбирает ровно один образ. Ограничение (3) обеспечивает, что разные вершины G_2 не могут быть сопоставлены одной и той же вершине G_1 . Ограничения (4), (5), (6) эквивалентны условию $w_{uvij} = x_{ui} \cdot x_{vj}$, но записаны в линейном виде.

2. Эволюционные алгоритмы

Для поиска приближенного решения задачи предлагается эволюционная стратегия $(1 + 1)$ -ES и генетический алгоритм, которые представляют собой эвристический метод оптимизации из класса эволюционных алгоритмов, основанный на адаптации и эволюции особей (пробных решений) [3, 7].

В качестве кодировки решений используется перестановка. К множеству вершин малого графа добавляются изолированные вершины, чтобы количество вершин в двух графах было одинаковым. Позиции перестановки соответствуют номерам вершин графа G_1 , а значения в позициях – номерам вершин графа G_2 .

При вычислении целевой функции сопоставляются пары вершин и осуществляется проверка наличия ребра в графах. Если ребро есть в обоих графах, то целевая функция увеличивается на 1, иначе – остаётся с прежним значением.

Эволюционная стратегия. Работа алгоритма начинается с особи-перестановки, сгенерированной случайным образом. На каждой итерации к текущей перестановке применяется один из операторов мутации, вносящих случайные изменения. Если действие оператора приводит к перестановке, соответствующей улучшению значения целевой функции по сравнению с текущим решением, то на следующей итерации используется построенная перестановка. В противном случае работа продолжается с текущей перестановкой. Схема данного алгоритма представлена ниже.

Эволюционная стратегия $(1 + 1)$ -ES.

1. Построить начальную перестановку π .
2. Пока не выполнен критерий остановки:
 1. Выбрать оператор мутации mut .

2. Построить $\pi' = mut(\pi)$.

3. Обновить π как лучшую по целевой функции перестановку между π и π' .

В $(1 + 1)$ -ES используются три различных оператора мутации: *swap*, *shift* и *shuffle* [7]. Первый оператор меняет местами два случайных элемента перестановки, второй оператор сдвигает случайный элемент перестановки в новую случайную позицию, третий же выбирает случайную подпоследовательность и случайным образом перемешивает элементы этой подпоследовательности. Для выбора оператора на каждой итерации алгоритма используется метод на основе обучения с подкреплением [1], основанного на целенаправленном обучении, которое производится путём взаимодействия условного агента с некоторой средой, в процессе которого направление действий агента обновляется на основе полученного опыта (см., рис. 1). Последовательность действий агента с вероятностью их выбора в некотором состоянии формируют стратегию агента. Целью агента является максимизация суммарного вознаграждения, которое он получает во время взаимодействия со средой.

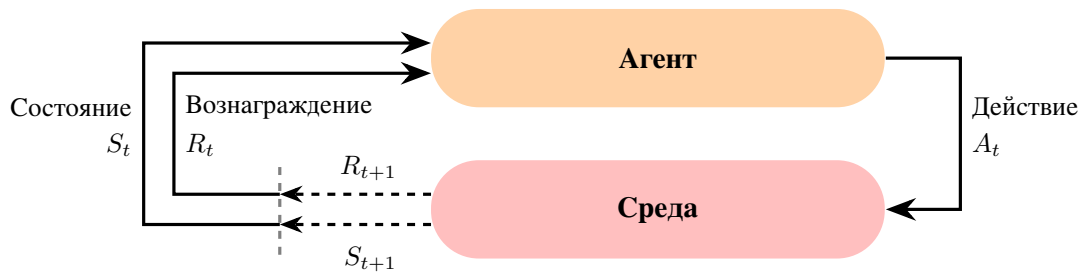


Рис. 1. Обучение с подкреплением

Существуют, например, такие алгоритмы обучения с подкреплением, как SARSA, Q-learning, метод на основе решения задачи о многоруком бандите и другие [1].

В данной работе используется метод обучения с подкреплением для выбора оператора мутаций. Адаптивный выбор операторов состоит из двух стадий:

1. Стадия обучения

Операторы: $op_1, op_2, op_3, \dots, op_k$.

- Операторы выбираются равновероятно.
- Обновление веса оператора op_i , $i = 1, \dots, k$:

$$w_i = \lambda \cdot \text{score} + (1 - \lambda) \cdot w_i,$$

где λ – параметр схемы, *score* – награда за улучшение (ухудшение) значения целевой функции.

2. Стадия подкрепления

- Оператор op_i , $i = 1, \dots, k$ выбирается с вероятностью:

$$p_i = \frac{w_i}{\sum_{j=1}^k w_j},$$

б) Обновление веса оператора op_i , $i = 1, \dots, k$:

$$w_i = \lambda \cdot score + (1 - \lambda) \cdot w_i.$$

Генетический алгоритм. Для решения задачи также предлагается генетический алгоритм со стационарной схемой воспроизведения, где решения как и в случае эволюционной стратегии кодируются перестановками.

В качестве оператора мутации используется оператор *swar*. Используется циклический кроссинговер [9], для которого выполняется свойство передачи генов в терминах наследования значений в позициях перестановки. В операторе селекции особи выбираются с помощью *s*-турнирной селекции (извлекается *s* случайных особей и из них выбирается лучшая по целевой функции).

Генетический алгоритм ГА

1. Построить начальную популяцию Π .
2. Пока не выполнен критерий остановки:
 1. Выбрать оператором селекции две родительских особи π_1 и π_2 .
 2. Применить к π_1 и π_2 оператор мутации.
 3. Применить к π_1 и π_2 оператор скрещивания и построить потомка π' .
 4. Заменить худшую особь в популяции на π' .

3. Экспериментальное исследование

Алгоритмы реализованы на языке Python с использованием встроенных библиотек. Эксперименты проводились на компьютере с процессором Intel Core i5-8400 2.8 GHz. Тесты проводились на серии задач с разной размерностью и плотностью графов, сгенерированных случайным образом на основе метода Эрдёша–Реньи. Размерности графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$: $|V_1| = 100$ и $|V_2| = 30$, $|V_1| = 200$ и $|V_2| = 60$, $|V_1| = 100$ и $|V_2| = 50$, $|V_1| = 200$ и $|V_2| = 100$ с плотностью 0.3, 0.5 и 0.8 для каждой из задач. Параметры генетического алгоритма: размер популяции – 50, размер турнира – 5. Для статистического анализа результатов на каждой задаче алгоритмы запускались 30 раз на 20 тысяч итераций.

Результаты экспериментов в терминах значений целевой функции $f(\cdot)$ представлены в таблицах 1, 2, 3, где в эволюционной стратегии используется только один оператор мутации, а в таблице 4 представлен результат работы эволюционной стратегии с адаптивной схемой выбора операторов. В обозначении задачи указывается число вершин в большом графе, малом графе и плотность графов.

В таблице 5 представлены средние отклонения от верхней границы (количество рёбер в малом графе) для всех исследуемых версий эволюционной стратегии и генетического алгоритма. Для оценки различий в целевой функции между значениями, найденными двумя $(1 + 1) - ES$: с оператором *swar* и адаптивной схемой, использовался тест Вилкоксона. Лучшие найденные значения для каждой задачи выделены жирным шрифтом, а символ «*» указывает на статистическую значимость отличий при уровне значимости $\alpha = 0.05$. Генетический алгоритм демонстрирует результаты сопоставимые с адаптивной эволюционной стратегией (результаты в среднем не более чем на 2 % хуже).

Таблица 1. Результаты (1 + 1)-ES с оператором swap

Задача	мин.	макс.	средн.
100/30_0.3	93	109	102,27
100/30_0.5	181	194	188,2
100/30_0.8	340	346	342,67
200/60_0.3	341	363	352,57
200/60_0.5	680	704	691,03
200/60_0.8	1329	1353	1340,83
100/50_0.3	230	252	243,03
100/50_0.5	465	485	473,8
100/50_0.8	911	933	923,13
200/100_0.3	824	859	840,73
200/100_0.5	1705	1760	1730,87
200/100_0.8	3566	3597	3581,83

Таблица 2. Результаты (1 + 1)-ES с оператором shift

Задача	мин.	макс.	средн.
100/30_0.3	73	87	81,4
100/30_0.5	157	172	164,07
100/30_0.8	321	333	327,4
200/60_0.3	254	289	274,4
200/60_0.5	577	609	593,3
200/60_0.8	1253	1288	1271,4
100/50_0.3	175	197	189,13
100/50_0.5	397	432	409,83
100/50_0.8	862	886	874,5
200/100_0.3	636	684	657,03
200/100_0.5	1487	1554	1512,5
200/100_0.8	3394	3446	3418,6

Таблица 3. Результаты $(1 + 1)$ -ES с оператором shuffle

Задача	мин.	макс.	средн.
100/30_0.3	85	94	89,2
100/30_0.5	166	183	175,4
100/30_0.8	328	337	333,6
200/60_0.3	288	305	296,9
200/60_0.5	604	641	624,17
200/60_0.8	1284	1311	1296,83
100/50_0.3	196	217	205,83
100/50_0.5	422	442	432,67
100/50_0.8	884	902	893
200/100_0.3	687	726	703,73
200/100_0.5	1551	1592	1571,6
200/100_0.8	3444	3480	3465,3

Таблица 4. Результаты адаптивной версии $(1 + 1)$ -ES

Задача	мин.	макс.	средн.
100/30_0.3	96	107	101,4
100/30_0.5	186	195	190,7
100/30_0.8	336	347	342,7
200/60_0.3	339	359	348,6
200/60_0.5	671	699	684,4
200/60_0.8	1327	1350	1340
100/50_0.3	227	254	242,9
100/50_0.5	464	484	472
100/50_0.8	910	933	924
200/100_0.3	810	847	830,7
200/100_0.5	1698	1735	1719
200/100_0.8	3562	3589	3575

Таблица 5. Отклонение среднего значения целевой функции от верхней границы (число рёбер в малом графе)

Задача	Верхняя граница	(1+1)-ES swap	(1+1)-ES shift	(1+1)-ES shuffle	(1+1)-ES adaptive	GA
100/30_0.3	130	21,33%	37,38%	31,38%	22,02%	21,28%
100/30_0.5	217	13,27%	24,39%	19,17%	12,12%*	13,21%
100/30_0.8	348	1,53%	5,92%	4,14%	1,52%	1,43%
200/60_0.3	531	33,60%*	48,32%	44,09%	34,36%	36,02%
200/60_0.5	885	21,92%*	32,96%	29,47%	22,67%	23,9%
200/60_0.8	1416	5,31%	10,21%	8,42%	5,40%	5,88%
100/50_0.3	367	33,78%	48,47%	43,92%	33,81%	33,99%
100/50_0.5	612	22,58%	33,03%	29,30%	22,88%	23,9%
100/50_0.8	980	5,80%	10,77%	8,88%	5,71%	5,71%
200/100_0.3	1485	43,39%*	55,76%	52,61%	44,06%	45,17%
200/100_0.5	2475	30,07%*	38,89%	36,50%	30,53%	31,49%
200/100_0.8	3960	9,55%*	13,67%	12,49%	9,73%	10,13%

По результатам, представленным в таблицах выше, можно сделать вывод, что оператор swap оказался эффективнее других операторов и показал схожие результаты с адаптивной схемой выбора операторов, но адаптивная схема имеет преимущество, так как может использовать несколько операторов, эффективность которых заранее неизвестна.

По сериям примеров разной размерности можно сделать следующие выводы:

1) на графах с плотностью 0.8 алгоритмы показывают более стабильные результаты с точки зрения целевой функции и это ожидаемо, так как, чем больше связей между вершинами в графе, тем больше шансов найти подграф, соответствующий малому графу;

2) на графах с плотностью 0.3 алгоритмы ведут себя менее стабильно, общее качество снижается;

3) при работе с более плотным графом алгоритмы демонстрируют меньший разброс значений и более высокие показатели качества решений.

Заключение

Проведено исследование оптимизационной версии задачи граф-подграф изоморфизма. Проанализированы вычислительная сложность и существующие методы, на основе чего разработана новая эволюционная стратегия с использованием набора операторов мутации и генетический алгоритм для указанной задачи. Реализована адаптивная схема выбора операторов, учитывающая их эффективность в процессе работы эволюционной стратегии. В результате экспериментального исследования и статистического анализа результатов показана применимость предложенных ал-

горитмов. Дальнейшие исследования могут быть направлены на исследование взвешенного варианта задачи и учёта дополнительных ограничений при сопоставлении вершин.

Благодарности

Работа выполнена при поддержке Математического Центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2025-349 от 29.04.2025.

Литература

1. Демченко М.В., Каширина И.Л., Фирюлина М.А. Использование методов обучения с подкреплением в задачах медицинской практики // Вестник ВГУ, серия: системный анализ и информационные технологии. 2022. № 1. С. 111–117.
2. Дидешко Д.С., Кравцов Г.Р., Захарова Ю.В. Об одном подходе к решению задачи граф-подграф изоморфизма // Молодёжь третьего тысячелетия: сборник научных статей XLIX региональной студенческой научно-практической конференции: в 3 ч. 2025. С. 224–227.
3. Еремеев А.В. Конспект лекций по курсу «Эволюционные алгоритмы». Омск, 2024. 75 стр.
4. Ильяшенко М.Б. Алгоритм нахождения граф-подграф изоморфизма для взвешенных графов и его применение // Радиоэлектроника. Информатика. Управление. № 1, 2007. С. 62–68.
5. Cho M., Lee J., Lee K. M. Reweighted random walks for graph matching // European conference on Computer vision. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. P. 492–505.
6. Haque M.N., Mathieson L., Moscato P. A memetic algorithm approach to network alignment: mapping the classification of mental disorders of DSM-IV with ICD-10 // Proceedings of the Genetic and Evolutionary Computation Conference. 2019. P. 258–265.
7. Holland J.H. Adaptation in Natural and Artificial Systems // An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, 1992. P. 89–118.
8. Klau G. W. A new graph-based method for pairwise global network alignment // BMC bioinformatics. 2009. V. 10. P. 1–9.
9. Reeves C. R. Genetic algorithms for the operations researcher // INFORMS Journal of Computing. 1997. V. 9, No. 3. P. 231–250.

EVOLUTIONARY ALGORITHMS FOR THE OPTIMIZATION VARIANT OF THE GRAPH-SUBGRAPH ISOMORPHISM PROBLEM

Yu.V. Zakharova^{1, 2}

Ph.D. (Phys.-Math.), Senior Researcher, e-mail: yzakharova@ofim.oscsbras.ru

V.P. Lobanov^{1, 2}

Ph.D. Student, e-mail: lobanov.vasily.pavlovich@gmail.com

G.R. Kravtsov³

Student, e-mail: grk55rus@mail.ru

¹Sobolev Institute of Mathematics, Omsk Department, Omsk, Russia

²Novosibirsk State University, Novosibirsk, Russia

³Omsk State Technical University, Omsk, Russia

Abstract. The article proposes evolutionary algorithms for the optimization version of the graph-subgraph isomorphism problem, which has applications in computer systems and bioinformatics. The genetic algorithm uses operators based on inheriting or mutating values at positions within permutations used to encode solutions. The evolutionary strategy employs an adaptive selection of operators. Experimental studies demonstrate the applicability of the developed algorithms.

Keywords: evolutionary strategy, genetic algorithm, isomorphism, experiment.

Дата поступления в редакцию: 26.01.2026

О ВОЗМОЖНОСТИ УПРАВЛЕНИЯ РОЛЕВОЙ ПОЛИТИКОЙ БЕЗОПАСНОСТИ ПОЛНОМОЧИЯМИ TAKE И GRANT

Д.М. Арушанян

магистрант, e-mail: arushanyandm@mail.ru

С.В. Усов

к.т.н., доцент, e-mail: raintower@mail.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. Предложена синтетическая модель, базирующаяся на ролевой модели безопасности, в качестве инструментов администрирования которой выступают полномочия take и grant, аналогичные эксплуатирующимся в одноимённой дискреционной модели. Выявлены виды ролевой иерархии, допускающие возможность такого синтеза.

Ключевые слова: компьютерная безопасность, ролевые модели разграничения доступа, иерархия ролей, модель Take-Grant, администрирование ролей.

Введение

Существующие модели, такие как дискреционный контроль доступа (DAC), мандатный контроль доступа (MAC) и ролевой контроль доступа (RBAC), получили широкое распространение. Несмотря на свои преимущества, каждая из этих моделей имеет заметные ограничения. DAC может быть ресурсоёмким, требуя детального управления разрешениями для множества субъектов и объектов. MAC, несмотря на высокую степень защиты, не обладает достаточной гибкостью для адаптации к различным операционным потребностям. RBAC, хотя и эффективен для упрощения управления разрешениями, может не соответствовать подробным и специфическим требованиям к доступу.

Чтобы преодолеть эти трудности, существует потребность в унифицированном подходе, который сочетает в себе сильные стороны традиционных моделей и в то же время смягчает их недостатки. Данная работа посвящена синтезу моделей контроля доступа, в частности, интеграции дискреционного подхода с принципами контроля доступа на основе ролей. Полученная в результате работы модель призвана обеспечить более гибкое решение для управления правами доступа.

Цель данной работы заключается в разработке ролевой модели разграничения доступа с механизмами, аналогичными тем, что присутствуют в модели Take-Grant.

1. Синтез гибридной модели разграничения доступа

Будем рассматривать базовую ролевую модель (RBAC) со множеством ролей R и общим множеством полномочий P [1]. Напомним, что каждой роли в RBAC со-

поставляется некоторое подмножество полномочий.

При необходимости, в зависимости от решаемой моделью задачи, на множестве ролей может быть введена иерархия, отражающая принцип наследования полномочий. Напомним три следующих подхода к организации структуры ролей:

- 1) строго таксономический листовой подход: при данном подходе полномочия назначаются листовым ролям, причём у каждой листовой роли в данном случае полномочия не пересекаются с остальными (каждое сопоставленное листовой роли полномочие уникально), а полномочия ролей-родителей формируются как объединение полномочий их потомков;
- 2) нетаксономический листовой подход: при данном подходе используются те же принципы, что и при строго таксономическом листовом подходе, но наборы полномочий листовых ролей могут пересекаться;
- 3) иерархический охватный подход: наследование в данном подходе происходит по тем же принципам, что были описаны выше, с тем только различием, что у родительских ролей допускается наличие полномочий, отсутствующих у их потомков.

Основная идея синтеза новой модели (в дальнейшем будем называть её RTGM – Role-based Take-Grant Model) состоит в добавлении к ролевой модели дополнительных администрирующих полномочий *take* и *grant*, которые будут аналогичны одноименным правам в дискреционной модели Take-Grant [2] для возможности альтернативного администрирования.

Введём для нашей модели следующий список требований, которым она должна соответствовать:

- модель базируется на ролевой модели (RBAC);
- набор полномочий, который принадлежит той или иной роли, модифицируется благодаря управляющим полномочиям *take* и *grant*, источником для которых служат одноимённые права доступа модели Take-Grant.

В дальнейшем, для удобства, будем обозначать представленные выше требования (*).

В общем случае компьютерная система будет представлена следующей четвёркой (P^*, R, Γ, G) , где:

- P^* – множество классических полномочий P , дополненное специальными администрирующими полномочиями *take*, *grant*, *create*, *delete* и *destroy* (далее будем обозначать эти полномочия соответственно t , g , c , l и d). Полномочие *take* позволяет заимствовать полномочие у целевой роли, *grant* – вручать полномочие целевой роли, *create* – создавать новую роль, *delete* – отзываться полномочия целевой роли, наконец, *destroy* – удалять целевую роль. Последние три полномочия введены на перспективу, как правило, мы будем работать с усечённой версией модели, в которой множество P дополняется лишь полномочиями *take* и *grant*.

В силу того, что RTGM берет механизмы администрирования из модели Take-Grant, в которой существуют понятия объектов, субъектов и прав доступа, под элементарным полномочием можно понимать пару $p = [o, \alpha]$, где o это объект, а α – набор прав доступа к нему. Администрирующие полномочия в

некоторых случаях могут аналогично определяться парой $t = [t, r]$ в случае с *take* и $g = [g, r]$ в случае с *grant*, где $\{t, g\}$ представляют собой способы взаимодействия, которые распространяются на роль r ;

- R – множество ролей, где каждая роль определяется набором полномочий, которые ей доступны;
- $\Gamma(R, E, \{t, g\})$ – tg-граф. Вершинами данного графа являются роли, и если роль r обладает полномочием t на роль r' , то в графе существует дуга от r к r' , подписанная полномочием t . Аналогичные дуги существуют и для полномочия g . Данный граф позаимствован из модели Take-Grant с тем лишь отличием, что в качестве вершин вместо объектов и субъектов выступают роли, а полномочия (за исключением *take* и *grant*) поставлены в соответствие не дугам, а вершинам графа.
- $G(R, T)$ – ориентированное дерево иерархии ролей, где R – это множество ролей системы, которые являются узлами дерева, T – множество дуг, направленных от родительских ролей к наследникам в рамках иерархической структуры ролей системы.

Множество пользователей и множество сеансов, которые присутствуют в классической ролевой модели, выведены за рамки данной работы, т. к. мы будем рассматривать модификацию иерархии ролей вне привязки к сеансам и пользователям; можно считать, что модификация происходит в рамках одного сеанса.

В силу наличия иерархии и графа иерархии введём отношение частичного порядка на множестве ролей:

$$F_{RR} : R \times R,$$

которое определяет иерархию ролей и задаёт оператор $\geq$, такой что если для $r_1, r_2 \in R$, $r_1 \geq r_2$, то r_1 находится в иерархии ролей выше чем r_2 .

Управление доступом в системе осуществляется в соответствии с отображением

$$F_{RP} : R \longrightarrow P^*,$$

где значением функции $F_{RP}(r)$ является множество полномочий, которыми обладает роль r .

Состояния подсистемы безопасности изменяются под воздействием команд пяти типов.

- Команда «Брать» ($\text{Take}(p, x, y)$) – роль x наделяется полномочием p , относящимся к роли y ($p \in F_{RP}(y)$). Команда осуществима, только если роль x обладает полномочием t ;
- Команда «Давать» ($\text{Grant}(p, x, y)$) – роль y наделяется полномочием p , относящимся к роли x ($p \in F_{RP}(x)$). Команда осуществима только если роль x обладает полномочием g ;
- Команда «Создать» – $\text{Create}(\alpha, x, y, \beta)$, где $\alpha \subseteq \{t, g\}$ – роль x создаёт роль y , обладающую множеством полномочий β , и получает полномочия α на неё ($\beta \subseteq F_{RP}(x)$). Команда осуществима, только если роль x обладает полномочием c ;
- Команда «Удалить» ($\text{Delete}(\beta, x, y)$) – роль x удаляет подмножество полномочий β роли y ($\beta \subseteq F_{RP}(y)$). Команда осуществима, только если роль x обладает полномочием l ;

- Команда «Уничтожить» ($\text{Destroy}(x, y)$) – роль x удаляет роль y . Команда осуществима, только если роль x обладает полномочием d .

Усечённая версия RTGM содержит только первые две команды, в дальнейшем (если не оговорено обратное) будет рассматриваться именно усечённая модель.

Возможность исполнения команд «Брать» и «Давать» зависит от существующей иерархии на множестве ролей, а также от конкретного способа реализации, что более подробно будет описано в последующих работах, освещающих отдельные случаи реализации модели. Но при наличии иерархии ролей исполнение никакой команды не может приводить к случаю нарушения целостности иерархии с точки зрения полномочий, а именно наличия у роли-потомка полномочия, отсутствующего у роли-родителя. Кроме того стоит отметить, что обе команды привязаны к ролям, т. е. команда «Брать» привязана к роли, у которой происходит похищение полномочия, а у команды «Давать» – к роли, которой мы передаём полномочие.

2. Согласование модели RTGM с иерархией ролей

Модель, представленная в работе, берет за основу ролевую модель, в связи с чем она оперирует полномочиями и ролями. В свою очередь, в ролевых моделях возможна организация ролей в рамках иерархической структуры, в связи с чем возникает необходимость установить, при каких видах иерархии возможна реализация модели RTGM.

Теорема 1. *Реализация RTGM с учётом требований (*) осуществима в случае:*

- 1) *отсутствия иерархии;*
- 2) *иерархического охватного подхода.*

и невозможна в случае:

- 1) *строго таксономического листового подхода;*
- 2) *нетаксономического листового подхода.*

Доказательство. Заметим, что сама передача полномочий не вызывает противоречий в базовой ролевой модели. Противоречия возникают при внедрении иерархии, а именно при ситуации, когда родительская роль получает полномочие, которого нет ни у одной дочерней роли. Такая ситуация противоречит организации листовых подходов. Из этого следует, что только наличие иерархии накладывает ограничение на внедрение и применение администрирующих полномочий t и g .

В графе иерархии либо существует одна ветвь, либо несколько. Случай с одной ветвью – вырожденный, т. к. в рамках листовых подходов у всех ролей единственной ветви будут одинаковые наборы полномочий, определённые полномочиями листовой вершины этой ветви. Поэтому, без ограничения общности, рассмотрим случай наличия хотя бы двух ветвей в рамках следующего примера.

Пусть дан произвольный граф иерархии G и tg-граф Γ (см. рисунок 1) со следующим распределением полномочий по ролям системы:

$$F_{RP}(r_{111}) = \{p_1\}; F_{RP}(r_{112}) = \{p_2\}; F_{RP}(r_{121}) = \{p_3, t\}; F_{RP}(r_{122}) = \{p_4\}.$$

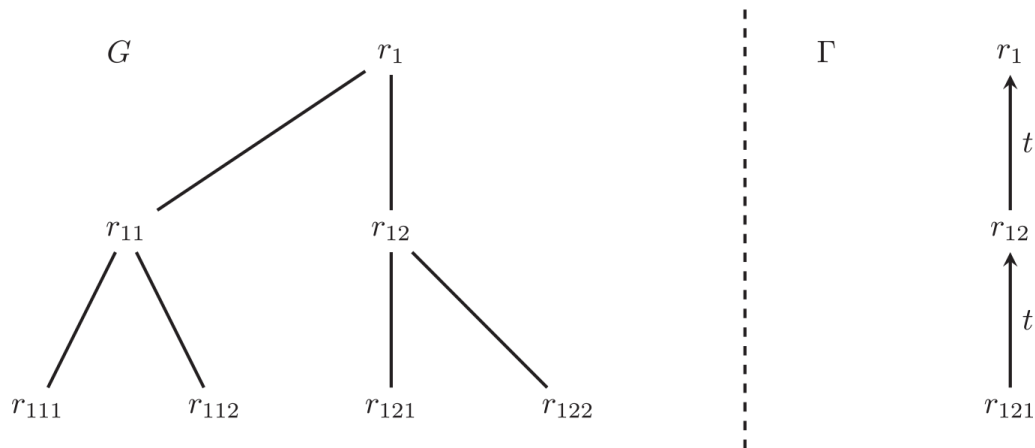


Рис. 1. Граф G_0 и Γ_0

Пусть мы используем строго таксономический или нетаксономический листовые подходы, которые подразумевают, что полномочия старших ролей являются объединением полномочий всех дочерних ролей, принадлежащих этой роли. Из этого следует, что

$$\{t\} \subseteq F_{RP}(r_{12}); \{t\} \subseteq F_{RP}(r_1);$$

Роль r_{12} может реализовать команду $\text{Take}(p_1, r_{12}, r_1)$, тем самым присвоить полномочие p_1 , которое принадлежит её родительской роли r_1 , тогда:

$$F_{RP}(r_{12}) = \{p_3, p_4, p_1, t\}.$$

Из этого следует, что

$$F_{RP}(r_{121}) \cup F_{RP}(r_{122}) \neq F_{RP}(r_{12}).$$

После применения $\text{Take}(p_1, r_{12}, r_1)$ роль r_{12} обладает полномочием p_1 , которое отсутствует у её потомков-листьев, что нарушает нетаксономический листовый подход и строго таксономический листовый подход, т. к. в данных подходах полномочия ролей-родителей являются объединением полномочий ролей-потомков, следовательно, множество полномочий всех листовых ролей должно совпадать с множеством полномочий роли-предка.

Приведённый пример демонстрирует ситуацию, когда в графе иерархии есть роль, которая может применить команду Take к роли, обладающей полномочием, которого нет у её потомков-листьев. В примере Take применяется ролью-похитителем к своему родителю, обладающему вторым наследником, но точно такой же эффект вызовет попытка применения команды Take к роли из другой ветви, если такое допустимо в рамках модели.

В результате исполнения команды нарушается строго таксономический листовый подход и нетаксономический листовый подход, но при этом подобная передача прав не создаёт противоречий при иерархически охватном подходе, т. к. в нем допускается наличие у родительской роли полномочия, которого нет у роли-потомка. ■

В последующих работах предполагается рассмотрение трёх возможных вариантов организации администрирования с помощью полномочий t и g :

- 1) команды Take и Grant используются в отношении всех полномочий, причём Take может быть применена только в отношении родительской роли, а Grant – только в отношении дочерней роли;
- 2) каждое полномочие t (или g) строго привязано к роли, по отношению к которой оно допускает применение соответствующей команды Take (или Grant), и согласовано с иерархией ролей;
- 3) внедрение команд Take и Grant рассматривается без использования иерархической структуры в рамках модели.

Литература

1. Ferraiolo D.F., Kuhn D.R. Role-Based Access Control // 15th National Computer Security Conference. October 1992. P. 554–563.
2. Jones A., Lipton R., Snyder L. A Linear Time Algorithm for Deciding Security // Proceedings of the 17th Annual Symposium on the Foundations of Computer Science. 1976.

ON THE POSSIBILITY OF MANAGING A ROLE-BASED SECURITY POLICY USING TAKE AND GRANT PERMISSIONS

D.M. Arushanyan

Master's Degree Student, e-mail: arushanyandm@mail.ru

S.V. Usov

Ph.D. (Techn.), Associate Professor, e-mail: raintower@mail.ru

Dostoevsky Omsk State University, Omsk, Russia

Abstract. In this paper, the possibility of synthesizing a hybrid access control model by integrating the role-based access control model with the discretionary Take-Grant model is considered. Role-based access control models augmented with administrative take and grant rights are examined. The resulting Role-based Take-Grant Model (RTGM) is formalized. The compatibility of the proposed model with various types of role hierarchies is analyzed. The feasibility of implementing the RTGM in systems without a role hierarchy and in those with a hierarchical coverage approach is proven. The contradictions arising from the application of the proposed model to strictly taxonomic and non-taxonomic leaf-based approaches are demonstrated.

Keywords: Role-based access control model, discretionary access control model, Take-Grant model, role hierarchy, administration rights.

Дата поступления в редакцию: 02.03.2026

УПРАВЛЕНИЕ СЛОЯМИ ГЕОПРОСТРАНСТВЕННЫХ ДАННЫХ ЧЕРЕЗ СПРАВОЧНИК ТИПОВ СУЩНОСТЕЙ

Т.Е. Болдовская¹

к.т.н., доцент, e-mail: boldovskaya73@gmail.com

М.И. Ремизов²

младший научный сотрудник, e-mail: remizov.maximus.maxim6@gmail.com

А.Е. Ветров²

младший научный сотрудник, e-mail: aleksandrvetrv838@gmail.com

¹Омский государственный технический университет, Омск, Россия

²Томский государственный университет, Томск, Россия

Аннотация. Рассматривается архитектура управления слоями геопространственных данных в веб-ориентированной геоинформационной системе, основанная на серверном справочнике типов сущностей. Предлагаемый подход устраняет жёсткую привязку конфигурации слоев к клиентскому коду и обеспечивает синхронизацию структуры отображаемых данных с актуальным состоянием базы данных. Особое внимание уделено серверной кластеризации точечных объектов, адаптивной загрузке данных по текущей области видимости карты и реактивной фильтрации слоев по типам сущностей. Описана многоуровневая схема доступа к данным, включающая OpenAPI-клиент, прикладной API-адаптер и репозиторий, инкапсулирующий агрегацию страниц, преобразование транспортных объектов и обработку ошибок. Показано, что сочетание серверной фильтрации, динамической конфигурации интерфейса и отмены устаревших запросов позволяет повысить производительность, управляемость и безопасность картографического приложения.

Ключевые слова: геоинформационные системы, геопространственные данные, серверная кластеризация, область видимости карты, справочник типов сущностей, динамическая конфигурация слоев, репозиторий.

Введение

Традиционные веб-приложения, работающие с многослойными картографическими представлениями, сталкиваются с рядом архитектурных ограничений при управлении динамическими источниками данных [1, 3]. В типичной архитектуре конфигурация слоев карты жёстко кодируется в исходном коде клиентской части в виде констант или конфигурационных объектов. При добавлении нового слоя разработчик вынужден производить изменения в исходном коде, что ведёт к полному циклу развёртывания (сборка, тестирование, деплой). Помимо этого страдает масштабируемость, поскольку при множественном добавлении слоев каждое изменение требует ручных правок кода.

Когда все слои карты являются постоянными значениями, находящимися на клиентской части приложения, в этом случае они становятся доступными всем пользователям без какой-либо дифференциации: «какие слои видит пользователь» либо полностью отсутствует, либо реализуется через условные конструкции в клиентском коде. Это является серьёзной проблемой безопасности: пользователь может отправить напрямую API-запрос на получение произвольного слоя или изменить переменную в инструментах разработчика в браузере. Поэтому контроль доступа к слоям необходимо выносить на серверную часть приложения, что позволяет отображать только разрешённые действия, логировать все попытки доступа на сервере и исключить возможность обхода проверок через манипуляции с клиентским кодом. При использовании справочника типов сущностей API возвращает только те слои, на которые у пользователя есть права, фильтруя их на основе таблицы слоев и текущей сессии пользователя.

В статье рассматривается комплексное решение, объединяющее серверную кластеризацию, адаптивную загрузку данных по области видимости карты и динамическое управление слоями через справочник типов сущностей.

1. Архитектура управления слоями через метаданные

Практическая реализация управления слоями через метаданные предполагает регистрацию слоя геоданных с указанием названия слоя, описания, проекции, прав доступа и других атрибутов. Геоданные разделяются на слои (классы, темы), при этом для каждого слоя создаётся отдельная директория, что позволяет логически разделить файлы различных слоев и управлять доступом к слоям на уровне директорий [2].

Критически важным является то, что метаданные (данные о данных), описывающие характеристики (содержание, объем, положение в пространстве, качество и т.д.) пространственных данных, содержат сведения о составе, статусе (актуальности и обновляемости), происхождении, местонахождении, качестве, форматах представления и условиях доступа. Это полностью соответствует концепции справочника типов сущностей, где каждый слой карты описывается набором атрибутов: уникальный идентификатор типа, семантическая метка для отображения, текстовое описание, флаг активности и уровень доступа. Применение этих принципов к картографическому контексту позволяет трактовать каждый слой карты как конфигурационный элемент с набором атрибутов, каждый из которых подчиняется формальной метамодели. Централизованное хранилище таких метаданных в БД обеспечивает единственный источник истины (single source of truth) для конфигурации слоев.

Концепция технологии ГИС заключается в создании многослойной электронной карты, опорный слой которой описывает географию территории. При этом критически важным является организация доступа к данным слоев через унифицированный интерфейс, независимый от внутренней структуры хранения. В свою очередь API-контракт действует как формальное соглашение между клиентом и сервером о структуре запросов и ответов, типах данных, ошибочных состояниях и семантике операций.

2. Серверная кластеризация геопространственных данных

Одной из ключевых проблем визуализации пространственных данных на интерактивных картах является необходимость отображения больших объемов точечных объектов при различных масштабах.

Традиционный подход – отправка и отрисовка каждого объекта как индивидуального маркера — неприемлем при масштабах в тысячи и более объектов, поскольку приводит к перекрытию маркеров, утрате читаемости карты и значительному росту нагрузки на сетевой канал и движок рендеринга.

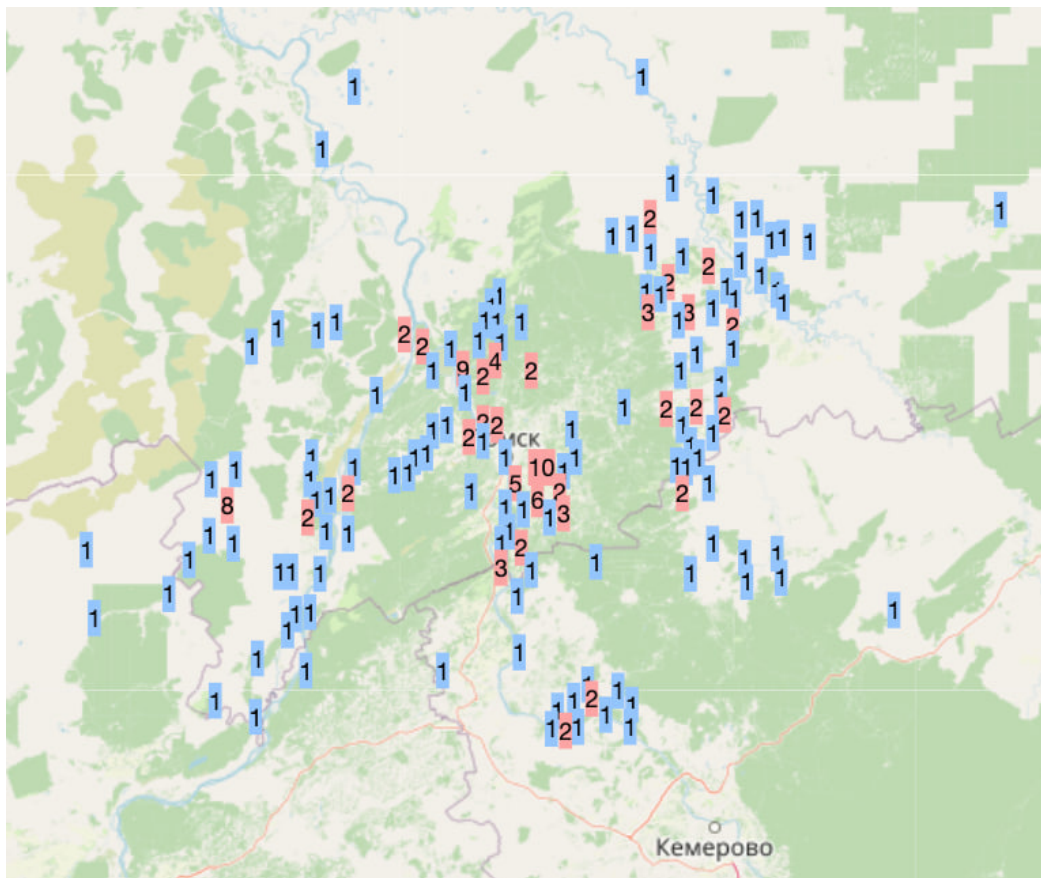


Рис. 1. Обзорное отображение геообъектов на карте

В рассматриваемой архитектуре кластеризация выполняется на серверной стороне, не затрагивая пользовательского интерфейса. Это принципиальное решение обусловлено несколькими факторами. Во-первых, серверная кластеризация позволяет оперировать полным набором данных, включая объекты за пределами текущей области видимости и обеспечивает корректное формирование кластеров на границах видимой области. Во-вторых, результаты кластеризации могут быть кэшированы на сервере для типичных комбинаций масштаба и координат, снижая вычислительную нагрузку при повторных запросах. В-третьих, пользователь получает уже агрегированные данные, что минимизирует объем передаваемого трафика.

Алгоритм серверной кластеризации принимает на вход два ключевых параметра: текущий уровень масштабирования и ограничивающий прямоугольник области видимости. Уровень масштабирования определяет пространственное разрешение кластеризации: при низких уровнях масштаба (обзор больших территорий) радиус кластеризации увеличивается, объединяя больше объектов в единый кластер; при высоких уровнях масштаба (детальный просмотр) радиус уменьшается, постепенно «раскрывая» кластеры до индивидуальных объектов. Ограничивающий прямоугольник задаётся четырьмя координатами: широта и долгота юго-западного угла, широта и долгота северо-восточного угла — что описывает прямоугольную область на земной поверхности, видимую пользователю в данный момент.

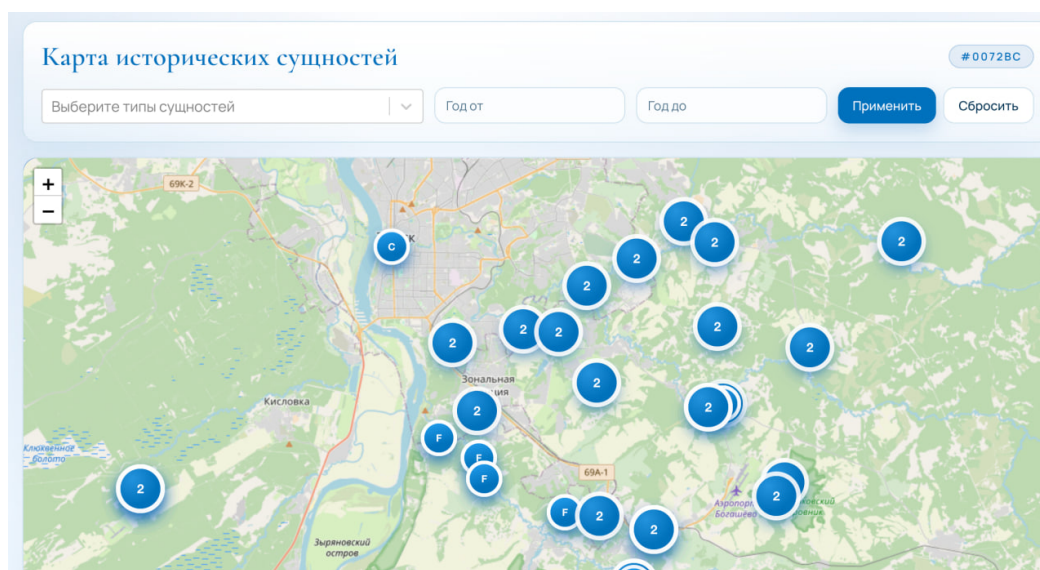


Рис. 2. Результат кластеризации при увеличении масштаба

Результат кластеризации возвращается в формате GeoJSON и содержит два типа геометрических объектов: индивидуальные точки и кластеры. Индивидуальная точка представляет собой единичный геообъект и несёт свойства, позволяющие идентифицировать и описать конкретную сущность. Кластер представляет собой группу пространственно близких объектов и содержит расширенный набор свойств: уникальный идентификатор кластера, количество объединённых объектов, а также коллекцию ссылок на входящие в кластер сущности. Геометрия кластера — это точка, соответствующая центроиду входящих объектов.

Такая дуальная структура ответа позволяет единообразно обрабатывать оба типа данных, визуализируя кластеры как агрегированные маркеры с числовым индикатором количества входящих объектов, а индивидуальные точки — как стандартные маркеры.

Дополнительно API поддерживает фильтрацию по типам существей: клиент передаёт массив идентификаторов типов, и сервер выполняет кластеризацию только для объектов, принадлежащих указанным типам. Пользователь получает возможность управлять видимыми «слоями» карты, включая и выключая отображение от-

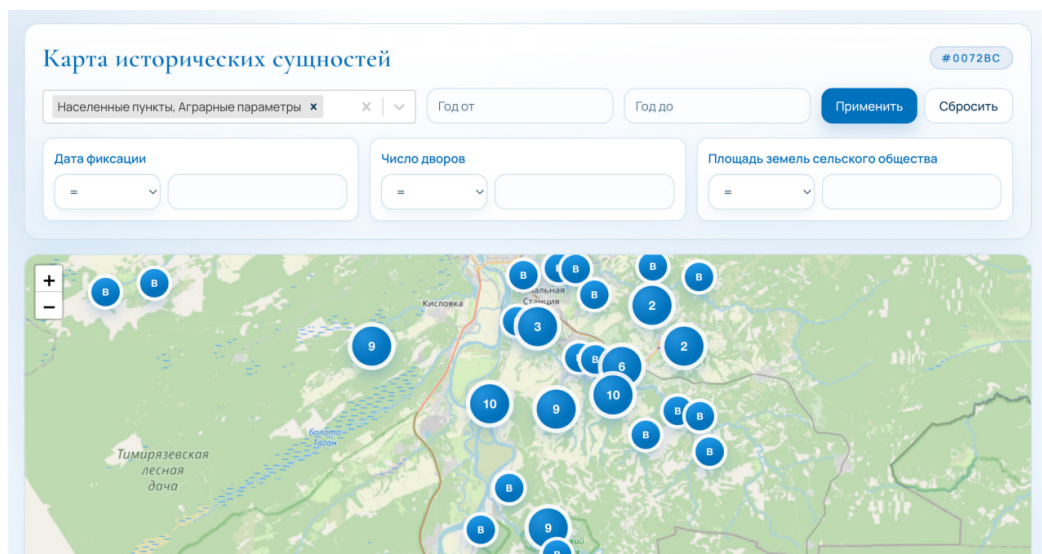


Рис. 3. Результат кластеризации при уменьшении масштаба

дельных классов объектов, при этом кластеризация автоматически пересчитывается с учётом текущего набора активных фильтров.

3. Адаптивная загрузка данных по области видимости

При реализации Информационно-аналитического ресурса «Православный ландшафт таёжной Сибири: акторы, институты, сети» вместо загрузки всего массива геоданных при инициализации карты реализована стратегия загрузки данных по запросу, привязанная к текущей области видимости карты [3]. В качестве примера, показывающего необходимость разработки алгоритмов визуализации объектов разного масштаба, является исследовательская задача по размещению объектов религиозной сети как связанных и коммуницирующих: деревень, имеющих устойчивую географическую локализацию, и таёжных старообрядческих монастырей, местоположение которых могло меняться в силу разных политических, социальных, экономических обстоятельств [4].

Работа алгоритма ресурса осуществляется следующим образом: каждое значимое пользовательское взаимодействие с картой — перемещение, изменение масштаба, поворот — генерирует событие окончания движения. Обработчик этого события извлекает из экземпляра карты актуальные параметры области видимости: центральные координаты (широта и долгота), текущий уровень масштабирования и координаты ограничивающего прямоугольника. Эти параметры сериализуются в строку URL-запроса и сохраняются в адресной строке браузера. Тем самым состояние карты синхронизируется с URL, что даёт два важных преимущества: возможность восстановления точного состояния карты при навигации по истории браузера (кнопки «назад» / «вперёд»), а также возможность передачи ссылки на конкретный вид карты между пользователями.

Обновление параметров в URL запускает реактивный механизм повторного за-

проса данных. Ключ запроса формируется из параметров области видимости и набора активных фильтров, поэтому при изменении любого из этих параметров система выполняет новый запрос к API. Для обеспечения визуальной непрерывности применяется стратегия «сохранения предыдущих данных»: пока новые данные загружаются, на карте продолжают отображаться данные предыдущего запроса, что исключает мерцание и пустую карту в процессе загрузки.

Запрос активируется условно: первоначальный запрос блокируется до момента инициализации карты и определения начальной области видимости. Только после того как карта полностью загружена и сообщила свои координаты через событие `onLoad`, система выполняет первый запрос данных, таким образом предотвращая отправку запросов с неопределёнными координатами.

Серверный API поддерживает страничную выдачу данных. Если количество объектов в запрошенной области превышает размер одной страницы, клиент выполняет параллельную загрузку всех оставшихся страниц. Механизм параллельной загрузки осуществляется через систему запросов. Первый запрос возвращает первичные данные (общее количество элементов, количество страниц, текущую страницу и ее размер). На основе этих метаданных формируется массив параллельных запросов для всех оставшихся страниц, которые выполняются одновременно. Результаты всех страниц объединяются: кластеры из всех страниц конкатенируются в единый массив, аналогично объединяются индивидуальные точки. Аналогично механизм объединения применяется и к индивидуальным точкам. В результате чего пользователь получает полный набор данных для текущей области видимости.

Валидация параметров запроса осуществляется на уровне маршрутизации через преобразование, ограничивающее его допустимым диапазоном (между минимальным и максимальным значениями). Координаты центра и ограничивающий прямоугольник являются опциональными параметрами и приобретают значение только после инициализации карты. Набор активных фильтров по типам сущностей также является опциональным, и при отсутствии интерпретируется сервером как «все типы».

4. Репозиторий и многоуровневый доступ к данным

В разработке приложений, требующих отделения бизнес-логики от деталей доступа к данным, широко применяются паттерны Repository и Data Access Object (DAO). Repository представляет собой уровень доступа к данным и выполняет двунаправленную передачу данных между слоями приложения, инкапсулирует логику извлечения данных в отдельном классе, предоставляя чистый, объектно-ориентированный интерфейс для манипуляции доменными сущностями. В рассматриваемой архитектуре доступ к данным организован через трёхуровневую иерархию. На нижнем уровне располагается автоматически сгенерированный API-клиент, полученный из формальной OpenAPI-спецификации серверного контракта. Этот клиент обеспечивает типобезопасное взаимодействие с REST-эндпоинтами и автоматическую сериализацию/десериализацию транспортных объектов (DTO). Средний уровень – прикладной API-адаптер – инкапсулирует сгенерированный клиент, добавляя возможность обработки сигналов отмены (`AbortSignal`) для прерыва-

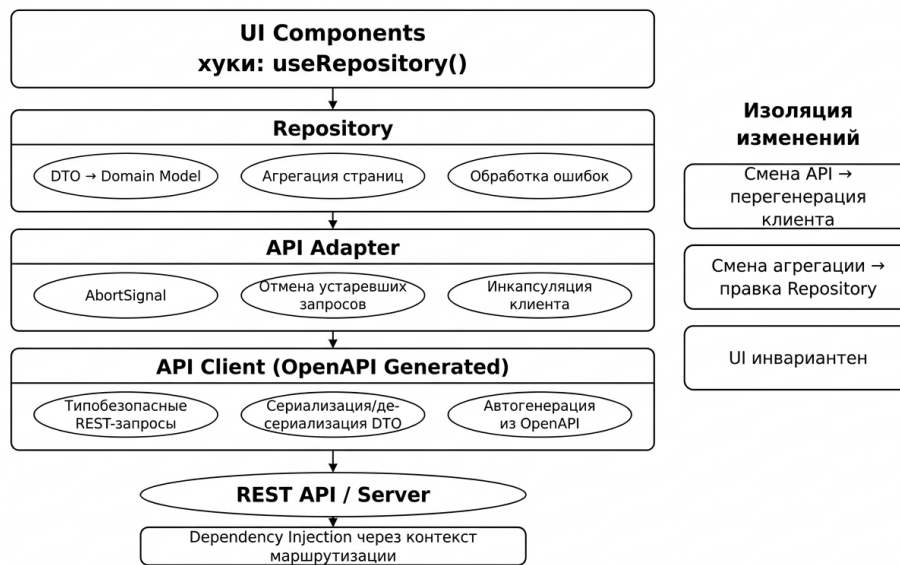


Рис. 4. Репозиторий и многоуровневый доступ к данным

ния устаревших запросов. Верхний уровень – репозиторий – реализует доменную логику: преобразование DTO в доменные модели, агрегацию страниц и обработку ошибок.

Многоуровневая структура, реализуемая в проекте, позволяет изолировать изменения: при обновлении серверного API достаточно регенерировать нижний уровень; при изменении бизнес-логики агрегации модифицируется только репозиторий. При этом компоненты пользовательского интерфейса остаются инвариантными к обоим видам изменений.

Применительно к справочнику типов сущностей паттерн Repository обеспечивает стабильный интерфейс к справочным данным независимо от выбранной СУБД или стратегии кэширования. Клиентское приложение всегда обращается к актуальному состоянию конфигурации без необходимости глубокого понимания серверной инфраструктуры.

Внедрение зависимостей (Dependency Injection) реализуется через контекст маршрутизации: на каждом уровне иерархии маршрутов создаются экземпляры репозитория и передаются вниз по дереву компонентов. Компоненты извлекают нужные репозитории с помощью специализированных хуков и обеспечивают инверсию контроля, упрощая тестирование.

5. Динамическая конфигурация слоев и безопасность

В клиентском приложении паттерн динамической конфигурации адаптирует поведение системы в реальном времени с использованием данных, полученных с сервера. При загрузке картографический интерфейс запрашивает у API справочник активных типов сущностей. Сервер возвращает массив объектов с уникальным идентификатором типа, семантической меткой и описанием. Полученные данные интер-

претируются и динамически конструируются в интерфейс управления слоями, позволяющий пользователю включать и отключать отображение определённых классов объектов.

Выбранные пользователем типы сущностей сериализуются в параметры URL наряду с параметрами области видимости карты. При каждом запросе данных на карту массив идентификаторов выбранных типов передаются серверу для выполнения операция фильтрации и кластеризации только объектов соответствующих типов. Данный запрос реализует реактивную связь: изменение набора активных фильтров мгновенно инициирует перезапрос данных с пересчитанной кластеризацией, и карта обновляется без перезагрузки страницы.

При добавлении нового типа сущности на серверной стороне клиентское приложение автоматически отобразит его в интерфейсе фильтрации без необходимости модификации клиентского кода. Это достигается за счёт того, что компонент фильтрации строится полностью на основе данных справочника, а не на основе постоянных значений.

Принцип информационного сокрытия в архитектуре приложений предполагает, что клиент должен иметь доступ только к той информации, необходимой для выполнения его непосредственных задач.

Информационная система ресурса содержит метаданные о слоях (название, описание, проекция, права доступа). Доступ к конкретным слоям регламентируется через систему прав доступа, исключая возможность несанкционированного получения данных. Система управляет доступом к слоям на уровне серверной фильтрации, обеспечивая изоляцию данных. Такая абстракция значительно сокращает поверхность атаки. Невозможно провести SQL-инъекцию или reconnaissance-атаку, основанную на методе перебора имён таблиц.

Дополнительным аспектом безопасности является механизм отмены устаревших запросов. При быстром изменении области видимости карты (последовательные операции масштабирования или панорамирования) предыдущие, ещё не завершённые запросы автоматически отменяются через сигнал отмены (AbortSignal), что снижает нагрузку на сервер.

Заключение

Рассмотренная архитектура демонстрирует комплексный подход к управлению геопространственными данными в интерактивных картографических системах. Серверная кластеризация с параметрическим управлением через уровень масштабирования и ограничивающий прямоугольник обеспечивает эффективную визуализацию больших объёмов данных при любом масштабе просмотра. Адаптивная загрузка данных по области видимости с механизмом параллельной постраничной загрузки, сохранением предыдущих данных и условной активацией запросов обеспечивает отзывчивый пользовательский интерфейс без избыточного сетевого трафика. Динамическое управление слоями через серверный справочник типов сущностей с полной синхронизацией фильтров и параметров карты через URL обеспечивает расширяемость системы без необходимости модификации клиентского кода. Совокупность этих подходов формирует архитектурный паттерн, применимый к

широкому классу геоинформационных систем, оперирующих динамическими, многослойными и объёмными наборами пространственных данных.

Благодарности

Данная работа выполнена при поддержке гранта Российского научного фонда № 23-78-10119.

Литература

1. Кухаренко Е.Л. Об опыте создания сервиса визуализации геоданных без программирования и его использовании в муниципальном образовании // Вестник СГУГиТ. 2025. Т. 30, № 5. С. 41–57.
2. Болдовская Т.Е., Ремизов М.И., Ветров А.Е. Систематизация и сохранение исторических источников в условиях цифровой эпохи // Математические структуры и моделирование. 2025. № 2 (74). С. 90–97.
3. Болдовская Т.Е., Гресь В.И. Информационно-аналитический ресурс «Православный ландшафт таежной Сибири: акторы, институты, сети»: архитектура системы, ключевые характеристики и интеграция исторических геоданных // Историческая информатика. 2025. № 1. С. 73–82. URL: https://nbpublish.com/library_read_article.php?id=74030 (дата обращения: 10.03.2026).
4. Дутчак Е., Ким Е., Буркун А. Крестьянская община и старообрядческий скит: формула притяжения // Православная культура вчера и сегодня / науч. ред. Е. Потехина, А. Кравецкий. Olsztyn: Uniwersytetu Warminsko-Mazurskiego w Olsztynie, 2015. С. 235–251.

MANAGING GEOSPATIAL DATA LAYERS VIA AN ENTITY TYPE DIRECTORY

T.E. Boldovskaya¹

Ph.D. (Techn.), Associate Professor, e-mail: boldovskaya73@gmail.com

M.I. Remizov²

Junior Scientist Researcher, e-mail: remizov.maximus.maxim6@gmail.com

A.E. Vetrov²

Junior Scientist Researcher, e-mail: aleksandrvetrv838@gmail.com

¹Omsk State Technical University, Omsk, Russia

²Tomsk State University, Tomsk, Russia

Abstract. The paper considers an architecture for managing geospatial data layers in a web-based geographic information system using a server-side entity type directory. The proposed approach removes hard-coded layer definitions from the client and synchronizes the displayed configuration with the current database state. Special attention is paid to server-side clustering of point objects, viewport-adaptive data loading, and reactive filtering by entity types. A multilayer data access scheme is described, including an OpenAPI client, an application API adapter, and a repository responsible for page aggregation, DTO-to-domain transformation, and error handling. The results show that combining server-side filtering, dynamic UI configuration, and cancellation of obsolete requests improves performance, maintainability, and security of the cartographic application.

Keywords: geographic information systems, geospatial data, server-side clustering, map viewport, entity type directory, dynamic layer configuration, repository.

Дата поступления в редакцию: 10.03.2026

ОКОННАЯ МОДЕЛЬ И РЕКУРРЕНТНЫЙ АЛГОРИТМ АГРЕГИРОВАНИЯ ТРАНЗАКЦИЙ В РАСПРЕДЕЛЁННОЙ БИЛЛИНГОВОЙ ПОДСИСТЕМЕ

И.Ф. Горкун

аспирант, e-mail: iliagorkun@yandex.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе предложена новая архитектура биллинговой подсистемы, основанная на оконной модели агрегирования транзакций и рекуррентном алгоритме обновления состояний. Проведён анализ исходного линейного алгоритма и показано, что при росте числа пользователей и транзакций он не удовлетворяет требованиям по срокам обработки и отказоустойчивости. Для устранения этих ограничений разработана формальная модель скользящего окна, описывающая поток транзакций и инкрементальное накопление агрегатов с постоянной вычислительной сложностью на событие. На основе этой модели построен двухфазный конечный автомат, реализующий транзакционность и идемпотентность в распределённой среде.

Ключевые слова: рекуррентный алгоритм, идемпотентность, конечный автомат, отказоустойчивость, горизонтальное масштабирование, распределённые системы, потоковая обработка данных.

Введение

Современные финансовые платформы неизбежно включают в свой состав компоненту биллинга, отвечающую за корректный расчёт, оформление и проведение платежей. От надёжности и точности работы этой подсистемы напрямую зависят как финансовые показатели предприятия, так и его репутационные риски. Сбой или некорректная работа биллинга может привести не только к существенным экономическим потерям, но и к юридическим последствиям, связанным с нарушением договорных и регуляторных обязательств.

В связи с этим к биллинговым системам предъявляются повышенные требования по качеству инженерной реализации, стабильности, отказоустойчивости и масштабируемости. Они должны гарантировать непрерывность обработки транзакций даже при частичных отказах инфраструктуры, а также обеспечивать соответствие бизнес-требованиям, связанным с ростом объёма операций и усложнением финансовых процессов.

Целью данного проекта являлась оптимизация существующего биллингового компонента, который столкнулся с проблемами стабильности и производительности при возросшей нагрузке. Основное внимание уделялось анализу текущих архи-

тектурных ограничений, повышению отказоустойчивости и переработке структуры системы в соответствии с современными принципами распределённых вычислений.

1. Анализ текущего алгоритма и требований

Биллинговая подсистема выполняет полный цикл расчётов для каждого пользователя в рамках банковского дня. Каждые сутки она собирает все транзакции, относящиеся к конкретному пользователю, агрегирует их по установленным правилам и формирует итоговый платёж. Этот процесс повторяется ежедневно и включает несколько стадий: выборку релевантных пользователей, обработку и суммирование транзакций, а затем выпуск платёжных документов в учётную систему. От корректности и своевременности выполнения этого цикла зависят точность расчётов и соблюдение финансовых обязательств.

Для оценки производительности и устойчивости системы необходимо формализовать её ключевые требования и ограничения. В данном разделе рассматривается исходный алгоритм работы биллинга, его временная модель и архитектурные характеристики. Особое внимание уделяется трём аспектам: соблюдению сроков завершения расчётного окна, масштабируемости при росте числа пользователей и транзакций, и устойчивости к отказам инфраструктуры.

1.1. Формализация требований

Определение 1 (Окно обработки). Пусть d — расчётный банковский день, для которого производится агрегирование транзакций и формирование платежей. Для этого дня определяются временные границы окна обработки:

$$s_d = 21:00 \text{ UTC } d - 1, \quad e_d = 03:30 \text{ UTC } d, \quad \Delta_{win} = e_d - s_d = 6,5.$$

Определение 2. Обозначим через $T_{complete}(u, d)$ момент времени, когда для пользователя u сформирован платёж в рамках расчётного окна дня d .

Требование R1 (срок). Для каждого пользователя u и дня d должно выполняться неравенство

$$T_{complete}(u, d) \leq e_d,$$

то есть процесс обработки данных и создания платежа должен завершаться до конца окна e_d .

Требование R2 (масштаб). Пусть U_d — число пользователей к выставлению счетов, N_d — число транзакций за день. Для всех U_d и N_d , не превышающих верхние границы U_{max} и N_{max} , система должна гарантировать корректное завершение обработки в пределах допустимого временного окна.

Требование R3 (отказоустойчивость). В кластере из M машин при одновременном выходе из строя не более $f < M$ узлов система должна обеспечивать продолжение работы и завершение обработки всех данных в пределах установленного временного окна.

SLO. Вероятностная гарантия завершения окна:

$$\mathbb{P}[T_{total}(d) \leq \Delta_{win}] \geq 1 - \varepsilon, \quad \varepsilon \leq 10^{-3}.$$

1.2. Изначальная архитектура

Определение 3 (Шаги исходного алгоритма). Процесс состоит из трёх стадий:

1. S1: выборка релевантных пользователей – выполняются сетевые и базовые запросы для определения активных аккаунтов;
2. S2: агрегация транзакций по пользователю – выполняются последовательные обращения к внешним системам и БД, требующие наибольших вычислительных и сетевых ресурсов;
3. S3: выпуск платежа – формирование итоговых платёжных сущностей и фиксация результатов с практически константной вычислительной сложностью на пользователя.

Модель сложности. Пусть U — число пользователей, а N — число транзакций. Введём аппроксимацию времени выполнения для каждой стадии в виде линейной модели:

$$T_1(U) = \alpha_1 U + \beta_1, \quad T_2(N) = \alpha_2 N + \beta_2, \quad T_3(U) = \alpha_3 U + \beta_3,$$

где $\alpha_i > 0$ — коэффициенты, характеризующие асимптотическую стоимость обработки одной единицы данных (пользователя или транзакции) на соответствующем этапе, а $\beta_i \geq 0$ — константные накладные затраты, не зависящие от размера входных данных (инициализация, синхронизация, сетевые задержки и т. п.). Совокупное время выполнения алгоритма задаётся суммой:

$$T_{\text{base}}(U, N) = T_1(U) + T_2(N) + T_3(U).$$

Лемма 1 (Неизбежность линейного роста). При фиксированной архитектуре исходного алгоритма время выполнения T_{base} увеличивается по мере роста U и N как минимум линейно.

Доказательство. Каждая транзакция просматривается не менее одного раза на стадии S2; каждый пользователь — как минимум на стадиях S1 и S3. $\square$

1.3. Эмпирическая калибровка

Пусть при текущем масштабе (U_0, N_0) наблюдалось:

$$T_1 \approx 1,5 \text{ ч}, \quad T_2 \approx 3 \text{ ч}, \quad T_3 \approx 10\text{--}15 \text{ мин.}$$

Тогда $T_{\text{base}}(U_0, N_0) \approx 5 \text{ ч.}$

Следствие 1 (Порог нарушения R1). Существует порог (U^*, N^*) , такой что для $U \geq U^*$ или $N \geq N^*$ выполняется

$$T_{\text{base}}(U, N) > \Delta_{\text{win}} - \delta_{\text{ops}},$$

где $\delta_{\text{ops}} > 0$ — операционный резерв (повторы, задержки). Тогда R1 нарушается для соответствующих сценариев.

1.4. Надёжность исходного запуска

Определение 4 (SPoF). Запуск на единственной машине задаёт одиночную точку отказа. При отказе в момент $t \in [s_d, e_d)$ процесс прерывается.

Теорема 1 (Уязвимость к отказам). В случае запуска на единственной машине ($M = 1$) вероятность нарушения требования R1 равна вероятности хотя бы одного сбоя в интервале $[s_d, e_d)$. Поскольку интенсивность отказов ненулевая, эта вероятность строго положительна.

Эскиз доказательства. Отказ единственной машины приводит к полной остановке процесса расчёта. Поскольку резервирование отсутствует, восстановление возможно только после ручного перезапуска, занимающего время, сопоставимое с длиной окна Δ_{win} , что делает выполнение требования R1 недостижимым при любом сбое. $\square$

1.5. Выводы

1. Ограничение окна Δ_{win} и линейный рост T_{base} по U, N приводят к риску систематического нарушения R1 при росте нагрузки.
2. Наличие SPoF в исходной схеме создаёт риск нарушения R3.

2. Новый алгоритм

2.1. Оптимизация процесса агрегации

Из анализа исходного алгоритма следует, что наибольшие издержки приходятся на этап агрегации данных, который выполняется в оперативной памяти одной машины и зависит от внешних систем. Это создаёт риски потери промежуточных результатов и ограничивает масштабируемость.

Перед тем как перейти к описанию новой архитектуры, введём формальную модель, описывающую процесс накопления и обновления агрегатов в потоковой системе. Нас интересует поведение алгоритма в рамках одного окна расчёта, когда транзакции поступают во времени и последовательно изменяют состояние агрегата. Такое представление позволяет строго определить корректность обновления, формализовать инварианты и далее построить отказоустойчивый алгоритм, реализующий эти свойства в распределённой среде. Для этого опишем поток событий, структуру окон и рекуррентный способ обновления агрегата.

2.2. Формальная модель скользящего окна

Определение 1 (Поток и порядок). Пусть $(\mathbb{T}, \leq)$ — множество моментов времени с частичным порядком, отражающим причинно-временные зависимости событий. Рассмотрим поток транзакций:

$$X = \{x_i\}_{i \in I}, \quad i < j \Rightarrow (t(x_i), id(x_i)) <_{lex} (t(x_j), id(x_j)),$$

где $t(x) \in \mathbb{T}$ — метка времени, а $id(x)$ — уникальный идентификатор транзакции. Так как несколько транзакций могут иметь одинаковое значение $t(x)$, для обеспече-

ния детерминированной последовательности обработки используется лексикографический порядок по паре (t, id) , аналогичный причинному порядку событий в рас-
пределённых системах [1].

Определение 2 (Окна). Фиксируем пользователя u . Его подмножество транзакций определяется как

$$X_u = \{x \in X : \text{user}(x) = u\},$$

где $\text{user} : X \rightarrow U$ — проекция, сопоставляющая каждой транзакции x идентификатор пользователя $u \in U$.

Пусть $\Delta > 0$ — фиксированная длина расчётного периода, а $\{b_k\}_{k \in \mathbb{Z}}$ — границы платёжных периодов, такие что $b_{k+1} - b_k = \Delta$. Окно для платежа k определяется как:

$$W_k = [b_k, b_{k+1}).$$

Требование «каждый платёж состоит из транзакций только своего периода» означает, что $\{W_k\}$ образует непересекающееся разбиение времени, аналогично оконным моделям, применяемым в потоковых системах обработки данных [2].

Определение 3 (Агрегат окна). Определим отображение

$$\varphi : X \rightarrow \mathbb{R}^m,$$

которое сопоставляет каждой транзакции x вектор её количественных характеристик (например, сумму, комиссию, налог и т.п.):

Агрегат для платежа k :

$$A_k = \sum_{x \in X_u : t(x) \in W_k} \varphi(x) \in \mathbb{R}^m.$$

Лемма 1 (Рекуррентное обновление в пределах окна). Фиксируем пользователя u и текущее окно $W_k = [b_k, b_{k+1})$. Пусть транзакции пользователя в этом окне образуют поток

$$X_{u,k} = \{x \in X_u : t(x) \in W_k\},$$

отсортированный по возрастанию метки времени $t(x)$. Определим агрегат после обработки первых n транзакций как

$$A_n(u, k) = \sum_{i=1}^n \varphi(x_i), \quad x_i \in X_{u,k}.$$

Тогда для каждой новой транзакции $x_{n+1} \in X_{u,k}$ выполняется рекуррентное соотношение:

$$A_{n+1}(u, k) = A_n(u, k) + \varphi(x_{n+1}).$$

Доказательство. Следует из аддитивности суммирования и линейности отображения φ :

$$\sum_{i=1}^{n+1} \varphi(x_i) = \sum_{i=1}^n \varphi(x_i) + \varphi(x_{n+1}) = A_n(u, k) + \varphi(x_{n+1}). \quad \square$$

Инварианты корректности

- *Линейность.* Для любых множеств $S, T \subseteq X_{u,k}$

$$\sum_{x \in S \cup T} \varphi(x) = \sum_{x \in S} \varphi(x) + \sum_{x \in T \setminus S} \varphi(x),$$

что гарантирует корректность пошагового накопления.

- *Детерминированность.* Порядок $(t(x), id(x))$ задаёт однозначную последовательность обновлений.
- *Аддитивность.* При поступлении новой транзакции агрегат изменяется только на $\varphi(x)$, без пересчёта предыдущих данных.

Персистентность рекуррентных состояний. Для отказоустойчивости значения A_k и служебная версия хранятся в таблице агрегатов с ключом $(userId, b_k)$, где $userId$ обозначает уникальный идентификатор пользователя, соответствующий u . Это позволяет восстанавливать состояние без хранения всего окна в оперативной памяти.

2.3. Транзакционность и идемпотентность через конечный автомат

Для формализации поведения распределённого расчётного процесса введём конечный автомат, описывающий переходы состояний агрегата при поступлении событий и фиксации транзакций.

Пусть $\mathcal{M} = (S, \Sigma, \delta, s_0)$ – конечный автомат, где:

- S – множество состояний агрегата расчёта для пары (пользователь, период);
- Σ – множество входных событий: поступление транзакции, закрытие окна, генерация платежа и т. д.;
- $s_0 \in S$ – начальное состояние (пустые агрегаты, версия 0);
- δ – двухфазная функция перехода ('BeforeTransaction' $\rightarrow$ 'InTransaction'), аналогичная классическим транзакционным моделям, описанным в [4].

Двухфазная семантика:

- *Подготовительная фаза* ('BeforeTransaction'):

$$pre : S \times \Sigma \rightarrow \{\text{commit, noop, reject}\} \times M,$$

где M — множество модификаторов $m : S \rightarrow S$.

- *Транзакционная фаза* ('InTransaction'):

$$tx : (S, m, k) \mapsto S,$$

выполняется атомарно в рамках БД-транзакции с дедупликацией по идемпотентному ключу k .

Идемпотентный ключ шага определяется как Идемпотентный ключ шага определяется как

$$k = Hash(u, b_k, id(x), \text{step}),$$

где $Hash(\cdot)$ – детерминированная хэш-функция, а параметр step обозначает тип выполняемой операции (например, AddTransaction, CreatePayout).

В хранилище поддерживается уникальный индекс по k , что обеспечивает равно-
однократность эффекта.

Функция применения события:

$$\text{Apply}(s, \sigma, k) = \begin{cases} s, & \text{pre}(s, \sigma) = (\text{noop}, _) \\ s, & \text{pre}(s, \sigma) = (\text{reject}, _) \\ m(s), & \text{pre}(s, \sigma) = (\text{commit}, m) \text{ и } \text{tx}(s, m, k) \text{ успешна.} \end{cases}$$

Гарантии:

- *Идемпотентность*: $\text{Apply}(s, \sigma, k) = \text{Apply}(\text{Apply}(s, \sigma, k), \sigma, k)$.
- *Транзакционность*: ‘InTransaction’ обеспечивает атомарность изменений состояния и регистрации ключа k .
- *Детерминизм*: модификатор m зависит только от (s, σ) ; побочные эффекты вынесены в идемпотентную фазу.
- *Линеаризуемость*: каждое состояние имеет версию $v(s) \in \mathbb{N}$; обновление версии гарантирует порядок.

2.4. Создание выплаты

Фиксируем окно $W_k = [b_k, b_{k+1})$. После завершения расчётного периода иници-
ируется событие закрытия окна:

$$\sigma_{\text{close}}(u, k) : \quad t_{\text{now}} \geq b_{k+1}.$$

Это событие обозначает момент, когда все транзакции, относящиеся к окну W_k ,
поступили и были обработаны в агрегатах пользователей. На этом этапе агрегаты
переходят из промежуточного состояния *Accumulating* в состояние *Aggregated*.

Множество кандидатов на выплату определяется как:

$$E_k = \{(u, k) : A_k(u) \text{ в состоянии } \text{Aggregated} \text{ и } A_k(u) > 0\},$$

Таким образом, в E_k попадают все пары (пользователь, окно), для которых накоп-
лены непустые результаты расчёта и агрегат готов к формированию платежа.

Каждый элемент $(u, k) \in E_k$ далее становится единицей работы для распреде-
лённого пула машин, обрабатывающего выплаты параллельно.

Распределение работ по машинам (устойчивое назначение). Пусть $h : (u, k) \rightarrow \{1, \dots, P\}$ – детерминированная хэш-функция (для балансиров-
ки), но источником истины о незавершённых работах является устойчивая
таблица кандидатов E_k в БД. Назначение пары (u, k) выполняется через *аренду* с
ограждающим токеном (fencing):

$$\text{Lease}(u, k) = (\text{owner}, \tau_{\text{exp}}, \nu),$$

где $\nu \in \mathbb{N}$ — версия аренды. Аренда захватывается атомарно при помощи операции
сравнения-и-обмена (CAS) в рамках транзакции. Если текущая запись отсутствует
или срок аренды истёк ($t_{\text{now}} > \tau_{\text{exp}}$), машина выполняет обновление:

$$(\text{owner} = s, \tau_{\text{exp}} = t_{\text{now}} + T_{\text{lease}}, \nu' = \nu + 1),$$

где T_{lease} – фиксированная длительность аренды (lease duration), задающая максимальное время владения задачей одной машиной. По истечении этого интервала право обработки автоматически может быть перехвачено другой машиной.

Любая последующая операция над парой (u, k) должна включать проверку текущей версии v (*fencing check*), что предотвращает возникновение *split-brain* ситуаций при конкурентном продлении аренды и гарантирует корректную эксклюзивность доступа.

При падении машины его аренда истекает, и другая машина может захватить работу без потери данных, так как агрегаты $A_k(u)$ и статусы хранятся в БД. Любая обработка выполняется атомарно и идемпотентна по ключу Идемпотентный ключ операции формирования выплаты определяется как

$$k_{\text{pay}} = \text{Hash}(u, b_k, \text{step} = \text{"createPayout"}, v(u, k)),$$

где:

- u – идентификатор пользователя;
- b_k – начало расчётного окна (идентификатор периода);
- step – тип выполняемого шага;
- $v(u, k)$ – версия состояния агрегата (u, k) , увеличивающаяся при каждом обновлении.
- $\text{Hash}(\cdot)$ – детерминированная хэш-функция.

Инварианты.

- (*Непотеря работ*) Множество кандидатов E_k хранится в БД; падение машины не удаляет и не скрывает (u, k) .
- (*Единоличное выполнение*) Fencing-версия v гарантирует, что только актуальный владелец может фиксировать изменения.
- (*Идемпотентность*) Уникальный индекс по k_{pay} обеспечивает равно-однократный эффект шага «создать выплату».
- (*Прогресс*) При наличии хотя бы одной живой машины и конечном T_{lease} все элементы E_k будут обработаны.

Формирование платежа. Пусть g – детерминированная функция построения платёжной сущности, определяющая преобразование агрегированных данных в итоговую запись о платеже:

$$P_k(u) = g(A_k(u), \text{meta}(u, k)),$$

где $A_k(u)$ — рассчитанный агрегат пользователя u за окно W_k , а $\text{meta}(u, k)$ – сопутствующие метаданные, включающие реквизиты пользователя, валюту, параметры выплат и дополнительные атрибуты расчёта. Функция g возвращает детерминированный результат – структуру платёжной записи, готовую к атомарной вставке или обновлению в базе данных. Идемпотентный ключ шага:

$$k_{\text{pay}} = \text{Hash}(u, b_k, \text{"createPayout"}, v), \quad pId = \text{Hash}(u, b_k, \text{"payoutId"}).$$

Здесь строковые литералы "createPayout" и "payoutId" выполняют роль семантических меток в хэш-функции, что соответствует принципам идемпотентной обработки событий и дедупликации сообщений, описанным в [3].

Двухфазное выполнение:

- **Подготовка:** вычисляем $P_k(u)$, проверяем инварианты, формируем модификатор m_{pay} .
- **Транзакция:** атомарно
 1. вставляем ключ k_{pay} в таблицу дедупликации;
 2. выполняем `upsert` платежа по (u, b_k) и pId ;
 3. переводим агрегат (u, k) в статус `Païd`, инкрементируем версию.

Гарантии корректности конца окна:

- Идемпотентность: повтор шага с тем же k_{pay} не изменяет состояние.
- Атомарность: все действия выполняются в одной транзакции.
- Прогресс: при наличии хотя бы одной живой машины и конечным времени аренды все $(u, k) \in E_k$ будут обработаны.
- Детерминизм: g и pId детерминированы по (u, k) .

Восстановление после сбоя. Если машина падает во время шага, то:

- либо транзакция не зафиксирована (аренда истекает, работа продолжается другой машиной);
- либо зафиксирована целиком (повтор приведёт к поор по k_{pay}).

Таким образом, сохраняются корректность и завершимость процесса.

2.5. Оценка ускорения и масштабирование

Новая схема заменяет полносканную агрегацию на инкрементальные обновления внутри окна (стоимость $O(1)$ на событие $\varphi(x)$) и распределяет работу по парам (u, k) на кластер из M машин с эффективной параллельностью $P_{\text{eff}} = M \cdot \eta$ (коэффициент утилизации $\eta \in (0, 1]$). Итого оценка времени выполнения:

$$T_{\text{new}}(U, N) \approx \frac{\alpha_1 U + \alpha_2 N + \alpha_3 U}{P_{\text{eff}}} + \delta_{\text{dist}},$$

где δ_{dist} — накладные расходы распределённой координации.

Для учёта неизбежной сериализации используем закон Амдала. Пусть $f_s \in [0, 1]$ — доля неизпараллеливаемых работ (координация, метаданные, публикации), тогда

$$T_{\text{new}}(U, N) \approx f_s \cdot T_{\text{base}}(U, N) + (1 - f_s) \cdot \frac{T_{\text{base}}(U, N)}{P_{\text{eff}}} + \delta_{\text{dist}}.$$

Численный пример (на базе эмпирики). При текущем масштабе наблюдалось:

$$T_1 \approx 1,5 \text{ ч}, \quad T_2 \approx 2\text{--}3 \text{ ч (возьмём } 2,5 \text{ ч)}, \quad T_3 \approx 0,2 \text{ ч},$$

т.е. $T_{\text{base}} \approx 4,2 \text{ ч}$.

Оптимистично-линейная оценка (без жёсткого сериализующего хвоста, $\delta_{\text{dist}} \approx 3 \text{ мин}$, $P_{\text{eff}} = \{6, 11, 16\}$):

$$T_{\text{new}} \in \{0,75, 0,43, 0,31\} \text{ ч} \Rightarrow \text{ускорение } S = \frac{T_{\text{base}}}{T_{\text{new}}} \in \{5,6, 9,7, 13,4\}.$$

Реалистичная оценка с законом Амдала ($\delta_{\text{dist}} \approx 3 \text{ мин}$):

$$f_s = 0,10, P_{\text{eff}} = \{8, 11, 16\} \Rightarrow T_{\text{new}} \approx \{0,94, 0,81, 0,71\} \text{ ч}, S \approx \{4,46, 5,16, 5,95\};$$

$$f_s = 0,05, P_{\text{eff}} = \{8, 11, 16\} \Rightarrow T_{\text{new}} \approx \{0,76, 0,62, 0,51\} \text{ ч}, S \approx \{5,54, 6,74, 8,25\}.$$

Вывод

В работе представлена и обоснована новая архитектура биллинга на основе оконной модели. Была описана формальная модель окон и рекуррентных агрегатов, доказана корректность их обновления и зафиксированы инварианты. Введён конечный автомат с двухфазными переходами (*подготовка* $\rightarrow$ *транзакция*) и идемпотентными ключами. Показаны свойства идемпотентности, атомарности и детерминизма, обеспечивающие линейризуемость по агрегату и устойчивость к повторам. Формализована процедура создания выплаты с шардированием и механизмом аренды (lease), гарантирующая прогресс при частичных отказах. Оценена вычислительная стоимость поддержки агрегатов как пропорциональная числу добавленных и удалённых элементов окна. Показано, что при увеличении числа машин пропускная способность системы растёт линейно до насыщения внешних ресурсов, что позволяет достигать целевых сроков обработки.

Литература

1. Lamport L. Time, Clocks, and the Ordering of Events in a Distributed System // *Communications of the ACM*. 1978. V. 21, No. 7. P. 558–565.
2. Carbone P., Katsifodimos A., Ewen S., Markl V., Haridi S., Tzoumas K. Apache Flink: Stream and Batch Processing in a Single Engine // *IEEE Data Engineering Bulletin*. 2015. V. 38 (4). P. 28–38.
3. Kleppmann M. Designing Data-Intensive Applications. O'Reilly Media, 2017. 616 p.
4. Gray J., Reuter A. Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993. 1070 p.

WINDOW-BASED MODEL AND RECURRENT TRANSACTION AGGREGATION ALGORITHM IN A DISTRIBUTED BILLING SUBSYSTEM

I.F. Gorkun

Ph.D. Student, e-mail: iliagorkun@yandex.ru

Dostoevsky Omsk State University, Omsk, Russia

Abstract. A new billing subsystem architecture is proposed, based on a windowed transaction aggregation model and a recurrent state-update algorithm. The analysis of the original linear algorithm shows that, with increasing numbers of users and transactions, it fails to meet timing and reliability requirements. To overcome these limitations, a formal sliding-window model is introduced, describing transaction streams and incremental aggregation with constant per-event computational cost. On this basis, a two-phase finite-state machine is constructed to ensure transactional and idempotent processing in a distributed environment.

Keywords: recurrent algorithm, idempotence, finite-state machine, fault tolerance, horizontal scalability, distributed systems, stream data processing.

Дата поступления в редакцию: 30.10.2025

ИСПОЛЬЗОВАНИЕ СИНТЕТИЧЕСКИХ ДАННЫХ И КО-СИМУЛЯЦИИ ANYLOGIC–PYTHON ДЛЯ РАЗРАБОТКИ GRU-MPC АЛГОРИТМОВ УПРАВЛЕНИЯ МИКРОКЛИМАТОМ ТЕПЛИЦ

Т.В. Костеннов

преподаватель, e-mail: timofey.kostennov@gmail.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе рассматривается подход к разработке алгоритмов предиктивного управления микроклиматом тепличных комплексов, основанный на совместном использовании синтетических данных и ко-симуляции AnyLogic–Python. На базе физико-математической модели теплицы формируется расширенный синтетический набор данных, описывающий различные сценарии погодных условий, управляющих воздействий и искажений измерений. На его основе выполняется предварительное обучение рекуррентной нейронной сети с управляемыми зависимостями (GRU) для краткосрочного прогноза параметров микроклимата, после чего модель дообучается на ограниченном объеме реальных производственных данных. Обученный прогнозирующий модуль интегрируется в контур предиктивного управления (GRU-MPC) и исследуется в среде ко-симуляции AnyLogic–Python. Показано, что использование синтетических данных позволяет повысить точность прогноза и устойчивость алгоритма управления в экстремальных режимах по сравнению с обучением исключительно на реальных данных, а ко-симуляция обеспечивает оценку эффективности и безопасности работы контроллера до проведения натурных испытаний.

Ключевые слова: микроклимат теплиц, предиктивное управление, рекуррентные нейронные сети, GRU, синтетические данные, цифровой двойник, ко-симуляция AnyLogic–Python.

Введение

Управление микроклиматом теплиц является одной из ключевых задач ресурсосберегающего растениеводства. Поддержание температуры, влажностного режима (RH/VPD), параметров грунта и концентрации CO_2 непосредственно влияет на урожайность, качество продукции и стрессоустойчивость растений, тогда как отопление, вентиляция, туманообразование и полив формируют значимую долю эксплуатационных энергозатрат [1–3]. Динамика тепличного микроклимата носит нелинейный и нестационарный характер и определяется совокупностью внешних и внутренних факторов, что снижает эффективность простых двухпозиционных регуляторов и затрудняет их настройку в условиях изменчивой и экстремальной погоды [4, 5]. В

связи с этим актуальны предиктивные методы управления, учитывающие динамику объекта, прогноз внешних возмущений, ограничения исполнительных механизмов и требования энергоэффективности [6–8].

Несмотря на результаты, достигнутые рекуррентными нейронными сетями (LSTM, GRU) при прогнозировании временных рядов, их применение в задачах управления теплицами ограничивается дефицитом репрезентативных производственных данных: реальная телеметрия содержит шумы, пропуски и изменения режимов эксплуатации, а экстремальные ситуации представлены недостаточно [9, 10]. Модели, обученные только на таких данных, нередко демонстрируют ухудшение обобщающей способности в непредставленных областях состояний, что повышает риски при переносе алгоритмов на объект.

Решением выступает использование цифровых двойников и синтетических данных [11]. Калиброванная физико-математическая модель теплицы в AnyLogic позволяет безопасно генерировать широкий спектр сценариев, включая экстремальные. На её основе формируется синтетический набор данных для предварительного обучения GRU-прогнозатора микроклимата, после чего модель дообучается на ограниченном реальном наборе данных, что снижает разрыв между синтетическими данными и практикой. Далее GRU-прогнозатор используется в MPC-контуре, реализованном в Python, причём AnyLogic функционирует как «виртуальная теплица» в среде ко-симуляции: на каждом шаге происходит обмен состояниями и управляющими воздействиями [12–14]. Это позволяет тестировать алгоритм в разнообразных условиях, сравнивать настройки MPC и анализировать влияние точности физической модели.

Цель работы – разработать и исследовать единый методический подход к разработке алгоритма предиктивного управления микроклиматом с комбинированным использованием синтетических и реальных данных и последующей проверкой GRU-MPC-контроллера в ко-симуляции AnyLogic–Python. Для этого описывается модель теплицы, разрабатывается метод генерации синтетического набора данных, проектируется архитектура GRU и процедура двухэтапного обучения, формулируется задача MPC с учётом качества регулирования, энергозатрат и ограничений, реализуется среда ко-симуляции и проводится сравнение предложенного подхода с базовыми регуляторами. Проверяемая гипотеза состоит в том, что совместное использование синтетических данных, реальных данных и ко-симуляции повышает точность прогноза, устойчивость и эффективность управления и снижает риски натурных испытаний по сравнению с подходом, основанным только на реальных данных [15].

1. Математическая модель теплицы

1.1. Структура объекта управления и основные переменные

Рассматривается отдельный тепличный модуль (одна или несколько одинаковых секций), описываемый сосредоточенной моделью с усреднёнными по объёму параметрами. Внутренний объём теплицы находится в тепло- и влагомассообмене с грунтом, прозрачным покрытием и внешней средой, а также с системами отопле-

ния, вентиляции и туманообразования [3, 4].

Основные *состояния* модели:

$$x = \begin{bmatrix} T_a & w_a & T_g & T_c \end{bmatrix}^T,$$

где T_a – температура воздуха, w_a – массовая концентрация водяного пара, T_g – эффективная температура верхнего слоя грунта, T_c – эффективная температура ограждений. При необходимости модель может быть расширена концентрацией C_{CO_2} .

Управляющие *воздействия*:

$$u = \begin{bmatrix} u_h & u_v & u_f & u_s \end{bmatrix}^T,$$

где u_h – относительная мощность отопления, u_v – степень вентиляции, u_f – интенсивность туманообразования/орошения, u_s – положение штор. При наличии досветки вводится u_l как дополнительное управление или возмущение.

Внешние *возмущения*:

$$d = \begin{bmatrix} T_{out} & w_{out} & G_{sol} & v_{wind} \end{bmatrix}^T,$$

где T_{out} , w_{out} – параметры наружного воздуха, G_{sol} – суммарная солнечная радиация, v_{wind} – скорость ветра.

1.2. Физико-математическая модель тепло- и влагопереноса

Модель задаётся системой ОДУ для балансов энергии и влаги в характерных подсистемах.

Баланс энергии для воздуха теплицы

Температура воздуха T_a описывается уравнением

$$C_a \frac{dT_a}{dt} = Q_{sol} + Q_{heat} + Q_{soil} + Q_{cover} + Q_{vent} + Q_{lat} + Q_{int}, \quad (1)$$

где $C_a = \rho_a c_{p,a} V_a$ – эффективная теплоёмкость воздушного объёма.

Основные слагаемые:

$$Q_{sol} = \tau_s(G_{sol}, u_s) G_{sol} A_c, \quad (2)$$

$$Q_{heat} = \eta_h F_h^{\max} u_h, \quad (3)$$

$$Q_{soil} = K_{ag}(T_g - T_a), \quad Q_{cover} = K_{ac}(T_c - T_a), \quad (4)$$

$$Q_{vent} = \rho_a c_{p,a} \dot{V}_{vent}(u_v, v_{wind})(T_{out} - T_a), \quad (5)$$

$$Q_{lat} = -L_v \dot{m}_{evap}, \quad (6)$$

где $\dot{V}_{vent}$ – расход вентиляционного воздуха, $\dot{m}_{evap}$ – массовая скорость испарения влаги. Q_{int} агрегирует внутренние тепловыделения.

Баланс влаги в воздухе

Влагосодержание воздуха w_a изменяется под действием испарения/транспирации, туманообразования, вентиляции и конденсации:

$$V_a \frac{dw_a}{dt} = \dot{m}_{\text{evap}}(T_a, w_a, \text{VPD}) + \dot{m}_{\text{fog}}(u_f) + \dot{m}_{\text{irr}} - \dot{m}_{\text{vent}}(u_v, v_{\text{wind}}) - \dot{m}_{\text{cond}}(T_c, w_a), \quad (7)$$

где потоки влаги задаются полуэмпирическими зависимостями, например

$$\dot{m}_{\text{evap}} = k_{\text{evap}} \text{VPD}(T_a, w_a) A_{\text{leaf}}. \quad (8)$$

Поток через вентиляцию:

$$\dot{m}_{\text{vent}} = \dot{V}_{\text{vent}}(u_v, v_{\text{wind}}) (w_a - w_{\text{out}}). \quad (9)$$

$\dot{m}_{\text{cond}}$ описывает конденсацию на внутренней поверхности покрытия.

Связь w_a с RH и VPD задаётся стандартными термодинамическими соотношениями:

$$\text{VPD} = e_{\text{sat}}(T_a) - e_v(T_a, w_a), \quad (10)$$

где e_{sat} – давление насыщенного пара, e_v – фактическое парциальное давление водяного пара.

Модель грунта и ограждений

Температура грунта T_g :

$$C_g \frac{dT_g}{dt} = K_{ag}(T_a - T_g) + Q_{g,\text{solar}} - Q_{g,\text{loss}}, \quad (11)$$

где $Q_{g,\text{solar}}$ – поглощённая радиация, $Q_{g,\text{loss}}$ – теплоотвод в грунт.

Температура ограждений T_c :

$$C_c \frac{dT_c}{dt} = K_{ac}(T_a - T_c) + K_{c,\text{out}}(T_{\text{out}} - T_c) + Q_{c,\text{rad}}, \quad (12)$$

где $K_{c,\text{out}}$ учитывает теплообмен с наружным воздухом и ветром, $Q_{c,\text{rad}}$ суммирует радиационный обмен.

Система (1)–(12) определяет динамику $x(t)$ при заданных $u(t)$ и $d(t)$. Численное интегрирование с шагом 5–15 минут реализовано в AnyLogic.

1.3. Учитываемые возмущения и управляющие воздействия

Модель чётко разделяет *управления* u и *возмущения* d , что важно для генерации синтетических данных и постановки МРС.

Управляющие воздействия $u_h, u_v, u_f, u_s \in [0, 1]$ описывают соответственно относительную мощность отопления, вентиляции, туманообразование/орошение и положение штор. При необходимости вводится управление досветкой u_l . Для оборудования учитываются ограничения по диапазону, скорости изменения и минимальному времени включения.

Возмущения $T_{\text{out}}(t)$, $w_{\text{out}}(t)$, $G_{\text{sol}}(t)$, $v_{\text{wind}}(t)$ задаются по данным метеостанции или с помощью синтетически сгенерированных профилей, в том числе с резкими изменениями для формирования экстремальных сценариев.

1.4. Кратко о калибровке и проверке модели

Параметры модели θ (коэффициенты тепло- и влагопереноса, эффективные теплоёмкости, параметры оборудования и др.) оцениваются по реальным данным через минимизацию невязки между моделью и измерениями:

$$J(\theta) = \sum_k \sum_{i \in \mathcal{Y}} w_i (y_i^{\text{mod}}(t_k, \theta) - y_i^{\text{meas}}(t_k))^2 \rightarrow \min_{\theta \in \Theta}, \quad (13)$$

где $\mathcal{Y}$ – набор наблюдаемых выходов (прежде всего T_a и RH/VPD), Θ – физически допустимая область параметров.

Используются стандартные комбинации градиентных и стохастических методов; детали не приводятся. После идентификации модель проверяется на независимых данных по RMSE, отсутствию выраженного смещения и корректному воспроизведению реакций на изменения погоды и управления. При достижении приемлемой точности модель рассматривается как цифровой двойник, достаточный для генерации синтетических данных и ко-симуляции AnyLogic–Python [5, 14].

2. Генерация синтетических данных

2.1. Сценарии погодных условий и внешних воздействий

Синтетический набор данных формируется путём прогонов цифрового двойника под набором погодных сценариев, т. е. временных профилей

$$T_{\text{out}}(t), w_{\text{out}}(t), G_{\text{sol}}(t), v_{\text{wind}}(t)$$

длительностью от нескольких суток до нескольких недель с шагом дискретизации 5–15 минут [1, 11].

Формируются несколько классов сценариев:

- штатные сезонные – типичные весенне-летние и осенние условия;
- холодные экстремальные – низкие температуры, быстрые похолодания;
- жаркие экстремальные – высокие температуры и радиация;
- переходные – быстрые изменения облачности, ветра и температуры.

Часть сценариев основана на реальных метеоданных со сглаживанием и масштабированием, часть – синтетическая (гармонические компоненты + шум). Контролируются диапазоны $T_{\text{out}}, w_{\text{out}}, G_{\text{sol}}, v_{\text{wind}}$, суточные циклы и базовые статистики.

2.2. Сценарии управления

Для каждого погодного профиля задаётся несколько вариантов управления, чтобы получить разнообразие траекторий:

- релейные регуляторы с гистерезисом по T_a и RH/VPD с разными порогами;
- простые пропорциональные/псевдо-PID-схемы по отклонению от уставок;
- сценарии с программным расписанием уставок (день/ночь, закалка и т.п.);
- возбуждающие (идентификационные) воздействия: ступенчатые и псевдослучайные изменения u_h, u_v, u_f в допустимых пределах.

Во всех случаях соблюдаются физические и агротехнические ограничения: $u_h, u_v, u_f, u_s \in [0, 1]$, учитываются минимальное время работы оборудования и недопустимость длительных экстремальных условий для растений.

2.3. Моделирование шумов, задержек и пропусков измерений

Чтобы синтетические данные были сопоставимы с реальными, к «идеальным» выходам модели добавляются искажения, имитирующие измерения:

- аддитивный шум с параметрами, соответствующими датчикам (температура, RH/VPD, T_g);
- медленный дрейф калибровки для части эпизодов;
- квантование (ограниченное разрешение измерений);
- задержка между фактическим состоянием и доступными измерениями;
- редкие пропуски и выбросы, обрабатываемые теми же методами фильтрации, что и для реальных данных.

После добавления шумов и пропусков к данным применяется единая технологическая процедура предобработки, аналогичный реальному (фильтрация, интерполяция, отсечение явных артефактов).

2.4. Структура синтетического набора данных и разделение train/val/test

На каждом шаге t_k сохраняются:

- текущие измеряемые переменные микроклимата (T_a , RH/VPD, при необходимости T_g и др.);
- управляющие воздействия $u_h(t_k)$, $u_v(t_k)$, $u_f(t_k)$, $u_s(t_k)$;
- внешние возмущения $T_{out}(t_k)$, $w_{out}(t_k)$, $G_{sol}(t_k)$, $v_{wind}(t_k)$;
- значения микроклимата на горизонте прогноза (1–3 часа вперёд).

Данные организованы по принципу «сценарий $\rightarrow$ временной ряд»; объём синтетического набора составляет десятки тысяч временных шагов (несколько виртуальных сезонов). Деление на train/val/test выполняется *по сценариям*, что исключает утечку информации:

- train – основная часть сценариев всех типов;
- val – отдельные сценарии (в т.ч. экстремальные) для подбора гиперпараметров;
- test – сценарии, полностью исключённые из обучения и валидации.

На основе этих временных рядов формируются скользящие окна для обучения GRU (2–6 часов истории и 1–3 часа прогноза).

2.5. Оценка реалистичности: сравнение синтетических и реальных данных

Реалистичность синтетического набора данных проверяется сравнением с историческими данными теплицы (1–2 сезона):

- сравниваются диапазоны и распределения T_a , RH/VPD, управляющих воздействий и внешних возмущений;
- анализируются усреднённые суточные профили и сезонные изменения;

- рассматриваются распределения приращений, автокорреляции и взаимные корреляции (например, между G_{sol} и ростом T_a);
- сравниваются амплитуда и характер шумов, частота выбросов и пропусков.

При необходимости корректируются параметры модели и генератора сценариев. Цель – не точное совпадение распределений, а физически правдоподобные и динамически корректные данные, расширяющие область сценариев за счёт экстремальных режимов.

3. Обучение GRU-модели на синтетических и реальных данных

3.1. Архитектура GRU-прогнозатора и формирование входных/выходных векторов

GRU-прогнозатор используется как динамическая модель микроклимата, связывающая историю состояний, управления и внешних возмущений с будущей траекторией. В дискретном времени $t_k = k\Delta t$:

$$y_k = \begin{bmatrix} T_a(k) \\ \text{VPD}(k) \end{bmatrix}, \quad u_k = \begin{bmatrix} u_h(k) \\ u_v(k) \\ u_f(k) \\ u_s(k) \end{bmatrix}, \quad d_k = \begin{bmatrix} T_{\text{out}}(k) \\ w_{\text{out}}(k) \\ G_{\text{sol}}(k) \\ v_{\text{wind}}(k) \end{bmatrix},$$

$$z_k = \begin{bmatrix} y_k \\ u_k \\ d_k \end{bmatrix}.$$

Для обучения строятся окна длиной L :

$$Z_k = \{z_{k-L+1}, \dots, z_k\}, \quad Y_k^{(\text{tar})} = \{y_{k+1}, \dots, y_{k+H}\},$$

где H – горизонт прогноза.

Используется один–два слоя GRU (32–128 скрытых нейронов) и полносвязный выходной слой (или маленькая MLP-«голова»), формирующий вектор

$$\hat{Y}_k = [\hat{y}_{k+1}, \dots, \hat{y}_{k+H}].$$

Входы стандартизируются (вычитание среднего и деление на стандартное отклонение). Модель реализует отображение

$$\hat{Y}_k = f_{\theta}(Z_k),$$

где θ – параметры сети. Обучение формулируется как задача многовыходной регрессии [9, 10].

3.2. Этап предварительного обучения (pre-training) на синтетическом наборе

На первом этапе GRU обучается на синтетическом наборе данных

$$\mathcal{D}_{\text{syn}} = \{(Z_k^{(\text{syn})}, Y_k^{(\text{syn})})\}$$

со случайной инициализацией параметров и разделением на train/val по сценариям. Используется Adam с шагом порядка 10^{-3} и L2-регуляризацией:

$$\mathcal{L}_{\text{syn}}(\theta) = \frac{1}{N_{\text{syn}}} \sum_k \sum_{h=1}^H \sum_j w_{j,h} \left(y_{k+h,j}^{(\text{syn})} - \hat{y}_{k+h,j} \right)^2 + \lambda \|\theta\|_2^2.$$

Качество контролируется по RMSE/MAE и R^2 для T_a и VPD, а остановка – по критерию ранней остановки на валидации. На этом этапе сеть выучивает базовую динамику теплицы, суточные и сезонные паттерны и реакции на экстремальные воздействия. Полученные параметры θ_{pre} используются как старт для дообучения.

3.3. Этап дообучения (fine-tuning) на ограниченном реальном наборе данных

На втором этапе GRU адаптируется к реальному объекту по набору данных

$$\mathcal{D}_{\text{real}} = \{(Z_k^{(\text{real})}, Y_k^{(\text{real})})\},$$

полученному из 1–2 сезонов реальной эксплуатации после предобработки. Архитектура и структура признаков сохраняются, параметры инициализируются из θ_{pre} , шаг обучения уменьшается, при необходимости часть слоёв «замораживается». Целевая функция:

$$\mathcal{L}_{\text{real}}(\theta) = \frac{1}{N_{\text{real}}} \sum_k \sum_{h=1}^H \sum_j w_{j,h} \left(y_{k+h,j}^{(\text{real})} - \hat{y}_{k+h,j} \right)^2 + \lambda_{\text{real}} \|\theta\|_2^2.$$

Используются те же метрики (RMSE, MAE, R^2), дополнительно анализируются смещение прогноза и ошибки в экстремальных режимах. Объём $\mathcal{D}_{\text{real}}$ существенно меньше объёма $\mathcal{D}_{\text{syn}}$, поэтому предварительное обучение играет роль структурной регуляризации.

3.4. Сравнение с моделью, обученной только на реальных данных

Для оценки влияния синтетических данных используется базовая модель, обученная только на $\mathcal{D}_{\text{real}}$ с той же архитектурой, признаками и процедурами нормировки. Обучение ведётся с нуля, с подбором гиперпараметров по валидации.

Сравнение двух моделей (**GRU (real-only)** и **GRU (syn+real)**) проводится на независимом реальном тестовом наборе по RMSE/MAE, R^2 , распределениям ошибок и поведению в экстремальных сценариях. Модель **GRU (syn+real)** показывает более низкие ошибки и более устойчивое качество прогнозирования, что подтверждает пользу предварительного обучения на синтетических данных и согласуется с результатами современных работ по предиктивному управлению микроклиматом теплиц [5, 15].

4. GRU-MPC и ко-симуляция AnyLogic–Python

4.1. Постановка задачи MPC и целевая функция

MPC формулируется с конечным горизонтом N и принципом скользящего горизонта. Управляемые переменные – температура воздуха T_a и показатель влажностного режима $z_{\text{hum}} = \text{VPD}$ с допустимыми диапазонами $[T_a^{\min}, T_a^{\max}]$, $[z_{\text{hum}}^{\min}, z_{\text{hum}}^{\max}]$.

Целевая функция:

$$J = \sum_{h=1}^N (\|T_{a,k+h} - T_{a,k+h}^{\text{ref}}\|_{q_T}^2 + \|z_{\text{hum},k+h} - z_{\text{hum},k+h}^{\text{ref}}\|_{q_H}^2) + \sum_{h=0}^{N-1} (\|u_{k+h}\|_{r_u}^2 + \|\Delta u_{k+h}\|_{r_{\Delta u}}^2), \quad (14)$$

где $u_{k+h} = [u_h, u_v, u_f, u_s]^T$, $\Delta u_{k+h} = u_{k+h} - u_{k+h-1}$. Первые слагаемые отвечают за качество микроклимата, вторые – за энергозатраты и плавность управления.

Ограничения по состоянию и управляющим воздействиям:

$$T_a^{\min} \leq T_{a,k+h} \leq T_a^{\max}, \quad z_{\text{hum}}^{\min} \leq z_{\text{hum},k+h} \leq z_{\text{hum}}^{\max}, \quad (15)$$

$$u^{\min} \leq u_{k+h} \leq u^{\max}, \quad \Delta u^{\min} \leq \Delta u_{k+h} \leq \Delta u^{\max}. \quad (16)$$

Внешние возмущения

$$d_{k+h} = [T_{\text{out}}, w_{\text{out}}, G_{\text{sol}}, v_{\text{wind}}]^T$$

на горизонте считаются известными (реальный прогноз или заранее заданный сценарий). На каждом шаге реализуется только первый элемент оптимальной траектории u_k^* , далее задача решается заново [6–8].

4.2. Интеграция GRU-прогнозов в контур MPC

Динамика теплицы в MPC аппроксимируется GRU-моделью. На шаге k контроллер имеет:

- текущий вектор y_k и последнее управление u_{k-1} ;
- прогноз возмущений $\{d_k, \dots, d_{k+N}\}$;
- историческое окно $\{z_{k-L+1}, \dots, z_k\}$.

Переменной оптимизации служит траектория $\mathbf{u} = (u_k, \dots, u_{k+N-1})$. Для заданной $\mathbf{u}$ GRU итеративно строит прогноз:

$$\hat{y}_{k+1} = f_{\theta}(Z_k, u_k, d_{k+1}), \quad (17)$$

далее окно сдвигается и получаются $\hat{y}_{k+2}, \dots, \hat{y}_{k+N}$. Это соответствует развёртке GRU по горизонту MPC с общими весами.

Поскольку f_{θ} дифференцируема по входам, задача MPC решается как нелинейная оптимизация с использованием автоматического дифференцирования: градиенты $\nabla_{\mathbf{u}} J$ вычисляются через backpropagation, оптимизация ведётся градиентными или квазиньютоновскими методами с учётом ограничений, по аналогии с работами по интеграции нейросетевых моделей в контуры предиктивного управления микроклиматом теплиц [5, 13, 14].

Для практической реализуемости выбирается умеренный горизонт N (например, 12–24 шагов), ограниченное число итераций оптимизатора и разумное начальное приближение (решение с предыдущего шага или базовая стратегия).

4.3. Архитектура ко-симуляции AnyLogic–Python и протокол обмена

Ко-симуляция AnyLogic–Python позволяет тестировать GRU-MPC на цифровом двойнике до натуральных испытаний. AnyLogic реализует модель теплицы и погодные сценарии, Python – GRU-прогнозатор и MPC.

При обмене с шагом Δt выполняются:

1. AnyLogic на шаге k вычисляет состояние x_k , формирует измерения y_k и возмущения d_k .
2. Через интерфейс (например, JSON по локальному TCP/HTTP) AnyLogic передаёт в Python пакет данных $\{y_k, u_{k-1}, d_k, \text{историческое окно}\}$.
3. Python обновляет буфер окна, при необходимости нормирует данные, решает задачу MPC и возвращает u_k^* .
4. AnyLogic применяет u_k^* на интервале $[t_k, t_{k+1})$ и интегрирует ОДУ до t_{k+1} .

Далее $k \leftarrow k + 1$ и цикл повторяется. Оба компонента работают синхронно: AnyLogic ждёт управления, Python – новых данных. Такая схема обеспечивает согласование «виртуального времени», удобный запуск серий сценариев и логирование всех переменных для анализа, и может быть реализована, в частности, с использованием библиотеки Alpyne для связи AnyLogic и Python [11, 12, 16].

4.4. Набор сценариев ко-симуляции (штатные и экстремальные режимы)

Для оценки эффективности и устойчивости GRU-MPC используется набор сценариев (см. раздел 2.), включающий:

- штатные суточные циклы (типичные дни с умеренными колебаниями);
- холодные экстремальные сценарии (длительные похолодания, ночные минимумы);
- жаркие сценарии (волны жары с высокой радиацией);
- быстрые переходы (резкая смена облачности, скачки температуры и ветра);
- комбинированные стресс-сценарии (неблагоприятная погода + ограничения по оборудованию).

Каждый сценарий проигрывается с GRU-MPC и с базовой стратегией (релейным или гистерезисным регулятором). Для каждого прогона рассчитываются интегральные ошибки (IAE/ISE по T_a и z_{hum}), суммарное энергопотребление, число переключений оборудования и показатели устойчивости (выходы за пределы, наличие/отсутствие колебаний). Это служит основой для сравнительного анализа в разделе 5..

5. Экспериментальные результаты

5.1. Качество прогноза: синтетические данные и реальные данные против только реальных данных

Качество GRU-прогнозатора оценивалось на независимой тестовой выборке реальных данных. Сравнивались две модели:

- **GRU (real-only)** – обучение только на реальном наборе данных;
- **GRU (syn+real)** – предварительное обучение на синтетических данных и дообучение на реальных данных.

Архитектура, схема признаков, нормировка и метрики совпадали. Оценка проводилась по RMSE и MAE для T_a и z_{hum} на горизонтах 1, 2 и 3 часа, коэффициенту R^2 и распределениям ошибок.

Модель **GRU (syn+real)** показывает более низкие RMSE/MAE и более высокий R^2 , особенно для больших горизонтов и в динамичных режимах (быстрая смена погоды, переходы день/ночь). В штатных условиях качество моделей сопоставимо, тогда как в экстремальных ситуациях преимущество синтетически предобученной модели выражено сильнее, в том числе за счёт уменьшения доли крупных ошибок. Смещение (bias) по температуре и VPD для **GRU (syn+real)** ближе к нулю, что важно для MPC.

5.2. Качество управления в ко-симуляции: GRU-MPC vs базовая стратегия

Каждый тестовый сценарий ко-симуляции проигрывается в двух вариантах:

- **GRU-MPC** – MPC с нейросетевым предсказателем **GRU (syn+real)**;
- **Базовая стратегия** – релейные или гистерезисные регуляторы с теми же диапазонами T_a и z_{hum} .

Для каждого варианта регистрируются траектории микроклимата, управления и возмущений. Оценка проводится по:

- интегральным ошибкам

$$\text{IAE}_T = \sum_k |T_a(k) - T_a^{\text{ref}}(k)|, \quad \text{ISE}_T = \sum_k (T_a(k) - T_a^{\text{ref}}(k))^2,$$

и аналогичным показателям по z_{hum} ;

- суммарному энергопотреблению

$$E = \sum_k (\alpha_h u_h(k) + \alpha_v u_v(k)) \Delta t;$$

- числу переключений основных исполнительных механизмов и распределению длительностей циклов работы.

GRU-MPC обеспечивает более точное поддержание T_a и z_{hum} в заданных диапазонах (меньше IAE/ISE) и более плавное управление с меньшим числом переключений. В большинстве сценариев наблюдается снижение энергозатрат, особенно при

нестабильной погоде, благодаря упреждающим действиям. В экстремальных сценариях GRU-MPC лучше удерживает микроклимат в допустимых границах при ограниченных ресурсах (мощность отопления, максимальная вентиляция). В отдельных случаях MPC может расходовать немного больше энергии, существенно улучшая качество микроклимата; такие компромиссы зависят от выбранных весов в функционале. Полученные эффекты находятся в одном ряду с результатами других работ по предиктивному управлению микроклиматом теплиц [5, 13, 15].

5.3. Анализ устойчивости и энергетической эффективности

Устойчивость и энергоэффективность анализировались по результатам длительных прогонов:

- переходы между режимами (день/ночь, смена уставок) сопровождаются сглаженными траекториями без значимых перерегулирований;
- при резких изменениях внешних условий контроллер реагирует упреждающе, уменьшая амплитуду отклонений микроклимата;
- на длительных интервалах не наблюдается самоподдерживающихся колебаний; система возвращается в допустимую область после сильных возмущений.

С точки зрения энергопотребления GRU-MPC:

- уменьшает число «чрезмерных» включений оборудования на полную мощность;
- распределяет действия по горизонту, позволяя работать ближе к нижним границам допустимых диапазонов в периоды высокого спроса на энергию;
- рационально комбинирует отопление, вентиляцию, шторы и туманообразование.

Эффект по энергосбережению зависит от веса энергетического слагаемого в целевой функции и конкретных агротехнических приоритетов.

5.4. Обсуждение ограничений и чувствительности к параметрам модели

Качество GRU-MPC ограничивается:

- точностью цифрового двойника: систематические ошибки модели (например, в оценке теплотерь или эффективности вентиляции) частично переносятся в синтетический набор данных и нейросетевой предсказатель; дообучение на реальных данных уменьшает, но не устраняет этот разрыв;
- выбором и охватом сценариев: недостаточное разнообразие ухудшает поведение в редких режимах, чрезмерное насыщение малореалистичными экстремумами может ухудшить качество в типичных условиях;
- настройкой гиперпараметров GRU и MPC (длина окна, горизонт, веса, ограничения по Δu), влияющих на баланс между упреждением, устойчивостью и вычислительной сложностью;
- качеством прогноза внешних возмущений: в ко-симуляции возмущения известны точно, в реальности – с погрешностью метеопрогноза, что требует робастных/стохастических модификаций MPC;

- доступными вычислительными ресурсами: решение MPC-задачи с нейросетевой динамикой в реальном времени требует оптимизированной реализации и, при необходимости, специализированного оборудования.

Тем не менее результаты ко-симуляции подтверждают, что использование синтетических данных и ко-симуляции AnyLogic–Python позволяет ускорить и безопасно провести разработку предиктивных алгоритмов управления микроклиматом теплиц и обеспечить выигрыш по качеству регулирования и энергоэффективности по сравнению с базовыми стратегиями.

Заключение

Разработана и исследована сквозная методика построения алгоритма предиктивного управления микроклиматом теплицы (GRU-MPC), основанная на совместном использовании синтетических и реальных данных и их интеграции в среду ко-симуляции AnyLogic–Python. Калиброванный цифровой двойник применён для генерации синтетической выборки по набору штатных и экстремальных сценариев погоды и управления, включая имитацию шумов и пропусков измерений. Предварительное обучение GRU-прогнозатора на синтетической выборке с последующим дообучением на реальных данных позволило снизить расхождение между моделью и объектом и повысить устойчивость прогноза в режимах, слабо представленных в реальной телеметрии. В ко-симуляционных экспериментах показано, что GRU-MPC по сравнению с базовыми двухпозиционными и гистерезисными регуляторами обеспечивает более точное удержание микроклимата в допустимых диапазонах, уменьшает интегральные ошибки регулирования и в большинстве сценариев снижает энергозатраты при одновременном сокращении числа переключений исполнительных механизмов.

Литература

1. Chen S. A Review of Environmental Control Strategies and Models for Modern Agricultural Greenhouses // *Sensors*. 2025. V. 25. P. 1388.
2. Боярцева В.К. Микроклимат теплиц. Справочник садовода. 2010. 420 с.
3. Байрамов Р.Б., Рыбакова Л.Е. Микроклимат теплиц на солнечном обогреве. Ашхабад: Ылым, 1983. 84 с.
4. Vanthoor B.H.E., Stanghellini C., van Henten E.J., de Visser P.H.B. A methodology for model-based greenhouse design: Part 1. A greenhouse climate model for a broad range of designs and climates // *Biosystems Engineering*. 2011. V. 110, No. 4. P. 363–377.
5. Mahmood F., Govindan R., Bermak A., Yang D., Al-Ansari T. Data-driven robust model predictive control for greenhouse temperature control and energy utilisation assessment // *Applied Energy*. 2023. V. 343. Art. 121190.
6. Rawlings J.B., Mayne D.Q., Diehl M. *Model Predictive Control: Theory, Computation, and Design*. Madison, WI: Nob Hill Publishing, 2017. 623 p.
7. Camacho E.F., Bordons C. *Model Predictive Control*. London: Springer-Verlag, 2004. 405 p.
8. Qin S.J., Badgwell T.A. A survey of industrial model predictive control technology // *Control Engineering Practice*. 2003. V. 11, No. 7. P. 733–764.

9. Liu Y., Li D., Wan S., Wang F., Dou W., Xu X., et al. A long short-term memory-based model for greenhouse climate prediction // *International Journal of Intelligent Systems*. 2022. V. 37, No. 1. P. 135–151.
10. Zhang G. et al. Predicting greenhouse air temperature using LSTM-AT // *Transactions of the Chinese Society of Agricultural Engineering*. 2024. V. 40, No. 18. P. 194–201.
11. Ariesen-Verschuur N., Verdouw C., Tekinerdogan B. Digital Twins in greenhouse horticulture: A review // *Computers and Electronics in Agriculture*. 2022. V. 199. Art. 107183.
12. McClellan A., Lorenzetti J., Pavone M., Farhat C. A physics-based digital twin for model predictive control of autonomous unmanned aerial vehicle landing // *Philosophical Transactions of the Royal Society A*. 2022. V. 380. Art. 20210204.
13. Fitz-Rodríguez E., Kacira M., Villarreal-Guerrero F., Giacomelli G.A., Linker R., Kubota C., Arbel A. Neural network predictive control in a naturally ventilated and fog cooled greenhouse // *Acta Horticulturae*. 2012. V. 952. P. 45–52.
14. Chen W.H. Data-driven robust optimization for greenhouse temperature and CO_2 concentration control // *Chemical Engineering Transactions*. 2020. V. 81. P. 671–676.
15. Soumo E.A., Calisti M., Polydoros A. Data-Driven Greenhouse Climate Regulation in Lettuce Cultivation Using BiLSTM and GRU Predictive Control. Preprint, 2025. URL: <https://arxiv.org/abs/2507.21669> (дата обращения: 16.12.2025).
16. Alpyne: A Python library for interactively running AnyLogic models. URL: <https://github.com/the-anylogic-company/Alpyne> (дата обращения: 16.12.2025).

SYNTHETIC DATA AND ANYLOGIC–PYTHON CO-SIMULATION FOR DEVELOPMENT OF GRU-MPC GREENHOUSE CLIMATE CONTROLLERS

T.V. Kostenov

Instructor, e-mail: timofey.kostenov@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. This paper presents an approach to the development of model predictive control (MPC) algorithms for greenhouse climate management based on synthetic data generation and AnyLogic–Python co-simulation. A physics-based greenhouse model is used to generate an extended synthetic dataset covering diverse weather conditions, control strategies and measurement distortions. This dataset is used to pre-train a gated recurrent unit (GRU) neural network for short-term climate prediction, which is then fine-tuned on a limited amount of real operational data. The trained forecasting module is integrated into a GRU-MPC controller and evaluated within an AnyLogic–Python co-simulation environment. The results demonstrate that combining synthetic and real data improves prediction accuracy and control robustness in extreme operating regimes compared to training on real-world data only, while co-simulation enables assessment of the controller's performance and safety prior to field deployment.

Keywords: greenhouse climate, model predictive control, recurrent neural networks, GRU, synthetic data, digital twin, AnyLogic–Python co-simulation.

Дата поступления в редакцию: 02.12.2025

АВТОМАТИЗИРОВАННОЕ ПРОГНОЗИРОВАНИЕ ПОСТПОЖАРНОЙ СУКЦЕССИИ ЛЕСОВ НА ОСНОВЕ МАРКОВСКИХ МОДЕЛЕЙ

О.А. Терентьева

к.т.н., доцент, e-mail: Anatole4ka@yandex.ru.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе представлена программная реализация марковской модели на Python для прогнозирования восстановления лесов после пожаров. Разработанное решение автоматизирует расчёт динамики сукцессии растительности с использованием уравнений Колмогорова–Чепмена. Программное приложение включает в себя численное решение уравнений модели, визуализацию результатов и интерактивный ввод параметров. Приведены примеры валидации модели на данных заповедника Утриш, демонстрирующие её точность и практическую применимость для планирования восстановительных мероприятий. Решение сокращает время расчётов в 3–5 раз по сравнению с ручными методами.

Ключевые слова: лесные пожары, прогнозирование, программное приложение, сукцессия, Марковский процесс.

Введение

Прогнозирование восстановления лесов после пожаров является важной задачей в экологическом мониторинге и управлении природными ресурсами. Современные методы автоматизации позволяют значительно ускорить и повысить точность таких прогнозов. Процесс постпожарного состояния леса включает в себя комплекс биологических, экологических и геохимических изменений, которые определяют дальнейшее развитие леса.

Одним из ключевых этапов восстановления является формирование пирогенных ландшафтов, которые включают в себя гибель растительности, изменение структуры почвы и являются стартовой точкой для сукцессии. Последний показатель носит положительный характер экологического последствия, так как пожар устраняет «больные» деревья и способствует обновлению экосистемы, а также создаёт мозаичность ландшафта, повышая биоразнообразие [1, 2].

Так, в работе учёных Черненко С.П. и Рогожиной Е.В. исследована произошедшая геохимическая трансформация почвенно-растительного покрова в разных местах, с разной интенсивностью пожара. Для изучения были взяты мхи, которые являются хорошими сорбентами и характеризуются максимально контрастными различиями в накоплении химических элементов в различных местах, подвергшихся пожару [3].

Максимальные накопления химических элементов в почвах обнаружались в местах интенсивного пожара. Был зафиксирован первый этап сукцессии для мхов в месте малоинтенсивного пожара, и установлен проход данного этапа сукцессии в месте максимального по интенсивности пожара.

В работе [4] предложена математическая модель, описывающая ситуацию успешности первого этапа сукцессии для мхов в зоне максимального по интенсивности пожара по сравнению с ходом сукцессии в зоне малоинтенсивного горения. В данной работе показано, что интенсивность пожара существенно влияет на скорость восстановления растительности. Однако существующие решения часто требуют ручного ввода параметров и не обеспечивают достаточной автоматизации расчётов [4, 5].

В данной работе рассмотрена марковская модель, учитывающая вероятностные переходы между стадиями сукцессии, и реализовано программное обеспечение для анализа динамики восстановления растительности. Модель была апробирована на данных заповедника Утриш, где в 2021–2023 гг. наблюдались пожары различной интенсивности.

1. Постановка задачи

Сукцессия моделируется на постпожарных участках посредством марковских процессов с непрерывным временем. Для каждого типа участка вводится конечное множество состояний, соответствующих стадиям восстановления растительности, и задаются интенсивности переходов между ними, а также задаётся матрица интенсивностей переходов Q .

Для участка с высокой интенсивностью пожара U_s :

$$Q_s = \begin{pmatrix} -\mu_0 & \mu_0 & 0 \\ 0 & -\mu_1 & \mu_1 \\ 0 & 0 & 0 \end{pmatrix},$$

где строки и столбцы соответствуют состояниям S_0, S_1, S_2 в указанном порядке.

Для участка с низкой интенсивностью пожара U_w состояния упорядочены как $S_0, S_{01}^w, S_1, S_{12}^w, S_2$:

$$Q_w = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 \\ 0 & -\mu_0 & \mu_0 & 0 & 0 \\ 0 & 0 & -\lambda_1 & \lambda_1 & 0 \\ 0 & 0 & 0 & -\mu_1 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Здесь:

- U_s – участок, подвергшийся пожару повышенной интенсивности;
- U_w – участок, подвергшийся соответственно пожару пониженной интенсивности;
- S_0 – начальная стадия сукцессии, послепожарная;

- S_1 – первая стадия, имевшая место через 1 год;
- S_2 – стадия сукцессии через 2 года, время наблюдений которой мы не фиксируем;
- S_{01}^w и S_{12}^w – стадии «подпитки» растений;
- λ_0, λ_1 – интенсивности перехода к стадиям подпитки.

Сукцессия в рассмотренной модели предполагает переход растительности из стадии S_i к стадии S_j с определённой вероятностью $P(ij)$, которые описываются как интенсивности, обозначаемые символом μ_i и понимаемым как среднее число событий в единицу времени, связанное с вероятностью $P(ij)$ перехода от стадии S_i к стадии S_j за время Δt :

$$P(ij)(\Delta t) = \mu_i \Delta t$$

при условии, что на отрезке времени Δt проявилось хотя бы одно событие из рассматриваемого потока событий [6].

Каждая стадия сукцессии наблюдается с некоторой вероятностью p_i . Динамика сукцессии описывается системой уравнений Колмогорова–Чепмена [7]:

$$\begin{aligned}\frac{dp_0}{dt} &= -\mu_0 p_0(t), \\ \frac{dp_1}{dt} &= -\mu_0 p_0(t) - \mu_1 p_1(t), \\ \frac{dp_2}{dt} &= -\mu_1 p_1(t),\end{aligned}$$

где $p_0(t), p_1(t), p_2(t)$ – вероятности нахождения в стадии S_0, S_1, S_2 , μ_0, μ_1 – интенсивности переходов, рассчитанные по данным [4].

2. Программная реализация

Программное приложение реализовано на языке программирования Python с использованием объектно-ориентированного подхода и состоит из трёх основных модулей:

- `markovmodel.py` – ядро расчётов. Функционал модуля реализует численный метод Рунге–Кутты 4-ого порядка для решения уравнений Колмогорова–Чепмена, а также поддерживает переменный шаг интегрирования для баланса точности и производительности;
- `visualization.py` – отрисовка графиков, а именно построение графиков динамики вероятностей;
- `interface.py` – пользовательский интерфейс для ввода данных и управления расчётами.

Схема взаимодействия описанных модулей представлена на рис. 1.

Алгоритм основных расчётов, которые реализованы и вычисляются программой, представлен на рис. 2.

Интерфейс приложения изображён на рис. 3.

Программа позволяет автоматизировать следующие этапы:

- ввод исходных параметров;

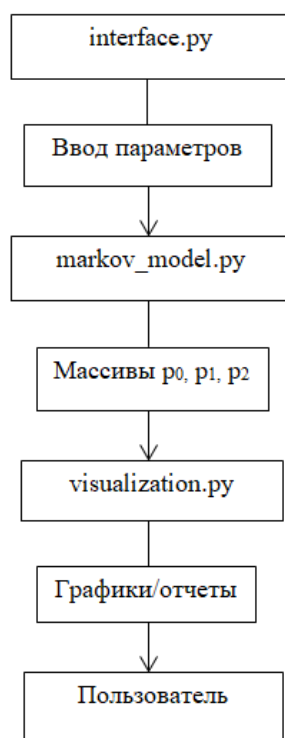


Рис. 1. Блок-схема взаимодействия модулей программы

- расчёт среднего времени этапа сукцессии T ;
- расчёт интенсивности μ_0 переходов растительности с начальной стадии S_0 в первую S_1 ;
- графическая иллюстрация вывода изменения вероятности состояний (стадий) сукцессии $p_0(t)$, $p_1(t)$, $p_2(t)$.

Программное приложение включает:

- окно для участков с высокой интенсивностью пожара – расчёт μ_0, μ_1 ;
- окно для участков с низкой интенсивностью – учёт стадий подпитки;
- окно сравнения – визуализация различий в динамике сукцессии.

Моделирование и вычислительные эксперименты апробированы автором на наблюдения постпожарной сукцессии в заповеднике Утриш (полуостров Абрау, Черноморское побережье России) пострадавшем от лесного пожара. Валидация модели проводилась по данным натурных наблюдений в заповеднике Утриш за 2021–2023 гг. Использовался поэтапный подход: интенсивности переходов оценивались методом максимального правдоподобия через минимизацию функции наблюдаемых и модельных вероятностей. Модель тестировалась на данных заповедника:

- для участков с высокой интенсивностью пожара U_s значения интенсивностей составили $\mu_0 = 0,35$, $\mu_1 = 0,25$;
- для U_w участков, подвергшихся соответственно пожару пониженной интенсивности $\mu_0 = 0,2$, $\mu_1 = 0,15$. Средняя ошибка между модельными и наблюдаемыми значениями не превысила 12,3 %, что подтверждает точность модели. Результаты

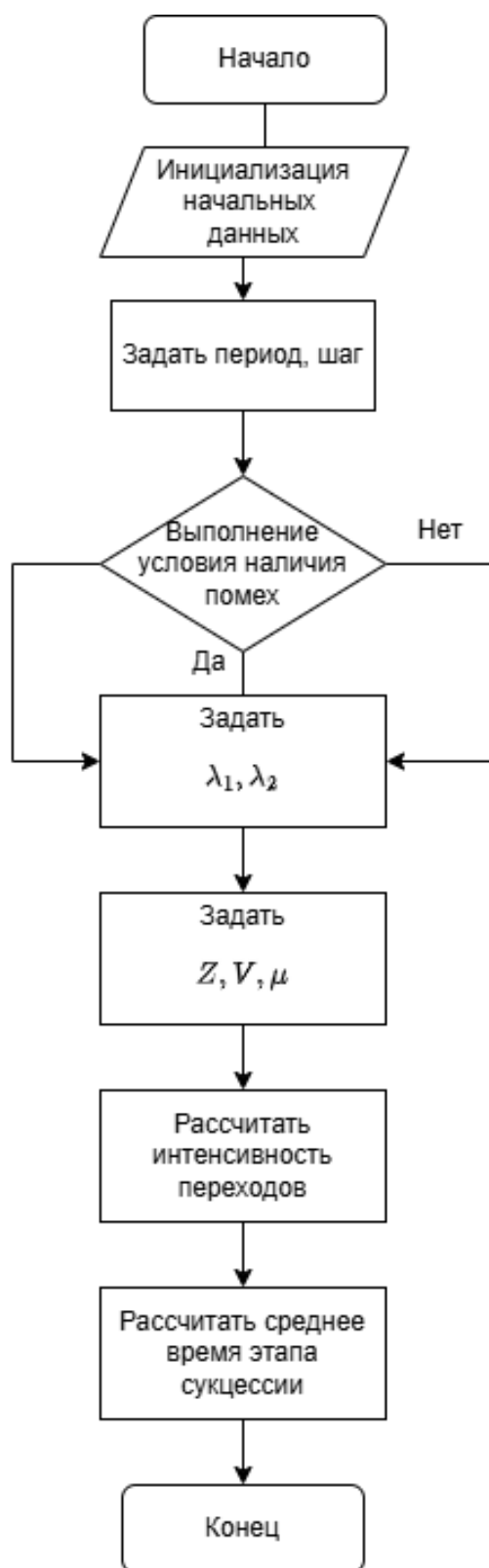


Рис. 2. Алгоритм программного приложения для расчётов постпожарной сукцессии

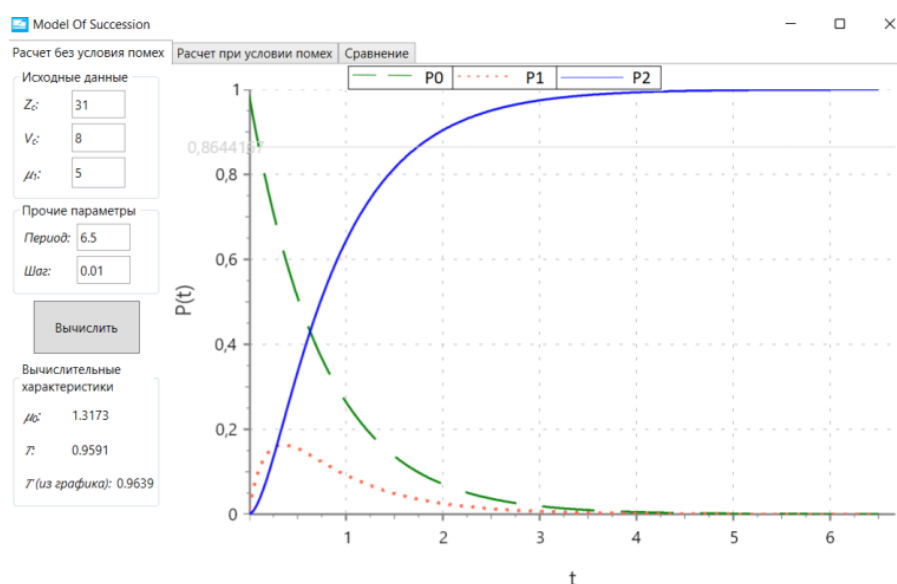


Рис. 3. Интерфейс программного приложения

эксперимента подробно представлены в работе [8].

Разработанная программная реализация марковской модели для прогнозирования восстановления лесов после пожаров демонстрирует значительные преимущества автоматизации экологического моделирования. Программа позволяет сократить время расчётов в 3–5 раз по сравнению с ручными методами. Например, расчёт динамики сукцессии для десятилетнего периода выполняется за 0,5–2 с, а ошибки человеческого фактора минимизируются.

Заключение

Было показано, что моделирование лесных пожаров представляет собой сложную задачу описания геохимической трансформации почвенно-растительного покрова поспирогенных зон различной интенсивности горения. Показана важность использования полученных данных постпожарной сукцессии для формирования пирогенных ландшафтов, которые служат стартовой точкой для сукцессии, определяя будущий облик экосистемы. Предложенная модель точнее учитывает стохастичность переходов по сравнению с детерминированными моделями, что подтверждается меньшими значениями средней ошибки. Разработанная программная реализация марковской модели позволяет автоматизировать прогнозирование восстановления лесов после пожаров. Программное обеспечение может быть использовано для планирования восстановительных мероприятий в заповедниках и лесных хозяйствах.

Дальнейшие исследования будут направлены на адаптацию модели для различных географических условий и интеграцию с системами дистанционного зондирования.

Благодарности

Автор выражает глубокую признательность своему научному руководителю – доктору физико-математических наук, профессору Александру Константиновичу Гуцу, память о котором автор навсегда сохранит. Его идеи, лежащие в основе данного исследования, неизменное внимание к работе, поддержка на всех этапах подготовки и человеческое тепло останутся для автора ориентиром на долгие годы. Автор искренне благодарен за мудрые советы, терпение и веру в успех, которые во многом определили научный путь и помогли сформироваться как исследователю. Светлая память об Учителе навсегда останется в сердце.

Литература

1. Немчинова А.В. Признаки стохастической детерминированности автогенной сукцессии лесных экосистем в марковских моделях // Компьютерные исследования и моделирование. 2016. Т. 8, № 2. С. 255–265.
2. Лесной кодекс Российской Федерации : от 04.12.2006 N 200-ФЗ (ред. от 29 декабря 2022 г.) // Доступ из СПС «КонсультантПлюс».
3. Черненко С.П., Рогожина Е.В., Керимзаде В.В. Активность каталазы и уреазы как индикатор пирогенной нарушенности почв (заповедник Утриш, полуостров Абрау) // Субтропическое и декоративное садоводство. 2022. № 82. С. 222–236.
4. Гуц А.К., Захарихина Л.В., Виницкая Е.А., Черненко С.П., Керимзаде В.В. Математическое моделирование геохимической трансформации почвенно-растительного покрова поспирогенных зон различной интенсивности горения // Математические структуры и моделирование. 2023. № 1 (65). С. 55–62.
5. Самороковский А. Ф. Использование теории марковских процессов при моделировании чрезвычайных ситуаций природного характера // Охрана, безопасность, связь. 2020. № 5–2. С. 81–86.
6. Венцель Е.С. Исследование операций. М.: Советское радио, 1972. 550 с.
7. Логофет Д.О. Марковские цепи как модели сукцессии: новые перспективы классической парадигмы // Лесоведение. 2010. № 2. С. 46–59.
8. Гуц А. К., Терентьева О.А. Программное приложение, визуализирующее расчёты постпожарной сукцессии мхов в заповеднике Утриш // Математические структуры и моделирование. 2023. № 2 (66). С. 80–90.

AUTOMATED FORECASTING OF POST-FIRE FOREST SUCCESSION BASED ON MARKOV MODELS

O.A. Terentyeva

Ph.D. (Eng.), Associate Professor, e-mail: Anatole4ka@yandex.ru

Dostoevsky Omsk State University, Omsk, Russia

Abstract. This paper presents a software implementation of a Markov model in Python for predicting forest recovery after fires. The developed solution automates the calculation of vegetation succession dynamics using the Kolmogorov–Chapman equations. The software application includes a numerical solution of the model equations, visualization of results, and interactive parameter input. Examples of model validation using data from the Utrish Nature Reserve are provided, demonstrating its accuracy and practical applicability for planning restoration measures. The solution reduces calculation time by 3-5 times compared to manual methods.

Keywords: forest fires, forecasting, software application, succession, Markov process.

Дата поступления в редакцию: 14.05.2023

СИСТЕМА ПРОФИЛИРОВАНИЯ UNITY3D С ФОРМИРОВАНИЕМ ДАТАСЕТОВ ДЛЯ ML-ОПТИМИЗАЦИИ

В.В. Фирстов

аспирант, e-mail: ckontro@gmail.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе рассматривается система автоматизированного сбора данных о производительности игровых сцен в среде Unity3D в режиме реального времени. Разработанное решение выполняет многократный прогон сцены с динамически изменяемыми параметрами рендеринга, объектов и окружения, фиксируя ключевые метрики производительности, включая частоту кадров, нагрузку на CPU и GPU, использование памяти и характеристики сцены. Полученные данные автоматически агрегируются в структурированный датасет, предназначенный для последующего обучения моделей машинного обучения. Предлагаемый подход позволяет перейти от ручного профилирования к анализу производительности, основанному на собранных данных, а также закладывает основу для построения предиктивных моделей, способных оценивать влияние параметров сцены на производительность еще на этапе разработки.

Ключевые слова: оптимизация производительности, разработка игр, Unity3D, машинное обучение, датасет.

Введение

Производительность игровых приложений в Unity3D зависит от множества взаимосвязанных параметров сцены, рендеринга и логики, что усложняет её анализ и оптимизацию. Традиционные методы профилирования основаны на ручном анализе отдельных конфигураций и плохо масштабируются при необходимости оценки большого числа вариантов настроек [1, 3].

Для систематического анализа производительности требуется автоматизированный сбор метрик при различных параметрах сцены. В данной работе предлагается система автоматического прогона Unity-сцен с динамически изменяемыми настройками и одновременным сбором ключевых метрик производительности в режиме реального времени. Полученные данные формируют структурированный датасет, предназначенный для последующего анализа и обучения моделей машинного обучения, что позволяет автоматизировать и упростить процесс оценки производительности.

Анализ существующих средств профилирования в среде Unity3D показывает, что в настоящее время отсутствуют решения, обеспечивающие автоматизирован-

ный многократный прогон игровых сцен с систематическим варьированием параметров рендеринга и контента, а также формированием структурированных датасетов, ориентированных на последующее использование в задачах машинного обучения.

Стандартные инструменты профилирования Unity предназначены преимущественно для интерактивного анализа отдельных конфигураций сцены и не предполагают ни воспроизводимого перебора параметров, ни накопления репрезентативных выборок данных, пригодных для статистического анализа и обучения моделей.

Таким образом, на момент проведения исследования отсутствуют программные системы, интегрированные в среду Unity3D, которые бы сочетали автоматизированное управление параметрами сцены, сбор многомерных метрик производительности в рантайме и формирование датасетов, ориентированных на задачи машинного обучения и предиктивной оптимизации.

1. Цель исследования

Основной целью настоящего исследования является разработка и описание системы автоматизированного сбора метрик производительности Unity-сцен в рантайме с динамически изменяемыми параметрами, обеспечивающей формирование структурированного датасета для последующего анализа и обучения моделей машинного обучения, направленных на оценку и прогнозирование производительности игровых сцен.

1.1. Параметры качества рендеринга

В ходе эксперимента автоматически изменялись следующие параметры сцены и рендеринга:

- качество теней;
- качество освещения;
- сглаживание;
- качество текстур;
- анизотропная фильтрация;
- смещение LOD;
- максимальный уровень LOD;
- отражения в реальном времени;
- мягкие частицы;
- качество скиннинга;
- трассировка частиц;
- качество шейдеров;
- количество пиксельных источников света;
- вертикальная синхронизация;
- дальность отрисовки;
- масштаб рендеринга;
- постобработка.

Все параметры принимали дискретные значения, соответствующие уровням качества Unity [2].

Дополнительно варьировались параметры, влияющие на вычислительную и контентную нагрузку:

- искусственная нагрузка на CPU;
- плотность контента сцены;
- активные группы контента.

Во время каждого прогона также фиксировались состояния среды выполнения:

- динамическое изменение разрешения;
- фактический масштаб рендеринга в рантайме;
- количество активных эффектов постобработки;
- состав активных эффектов.

1.2. Собираемые метрики производительности

В процессе прогона сцены с заданным интервалом выполнялись замеры следующих параметров:

- частота кадров (FPS);
- время кадра (мс);
- минимальное, максимальное и среднее значение FPS;
- среднее время кадра;
- средняя нагрузка на CPU;
- средняя нагрузка на GPU;
- время CPU-кадра (мс);
- время GPU-кадра (мс);
- использования памяти приложением;
- выделенная память;
- зарезервированная память;
- использование Mono-памяти;
- размер Mono-кучи;
- использование видеопамяти;
- количество draw calls;
- количество batches;
- количество треугольников;
- количество SetPass вызовов.

2. Ход работы

В данном разделе описывается процесс проведения исследования, включающий разработку тестовой сцены, организацию автоматизированного прогона и сбор метрик производительности.

2.1. Описание экспериментальной сцены

В рамках исследования была разработана тестовая сцена (рис. 1) в среде Unity3D, предназначенная для моделирования типовой игровой нагрузки и проведения серии автоматизированных прогонов с изменяемыми параметрами [8, 9]. Сцена проектировалась таким образом, чтобы включать основные компоненты, оказывающие влияние на производительность современных игровых приложений.

В состав сцены входили статические и динамические трёхмерные объекты различной сложности, использующие материалы с различными настройками шейдеров и текстур. Часть объектов обладала анимациями, реализованными с использованием стандартных средств анимационной системы Unity, что позволяло моделировать нагрузку, связанную с обновлением трансформаций и скелетной анимацией.

Для создания визуальной и вычислительной нагрузки в сцене использовались системы частиц, отличающиеся количеством эмитируемых элементов, типами рендеринга и наличием взаимодействия с геометрией сцены. Параметры частиц динамически изменялись в процессе эксперимента, что позволяло оценить их влияние на производительность.

Сцена также включала физические объекты с коллайдерами и компонентами физического моделирования, обеспечивающими расчёт столкновений и динамики в реальном времени. Количество активных физических объектов и сложность взаимодействий варьировались в рамках отдельных конфигураций.

Освещение сцены формировалось с использованием нескольких источников света различных типов, включая направленные, точечные и источники с динамическими тенями. В зависимости от конфигурации изменялись параметры освещения, качество теней и количество активных источников света, что оказывало влияние как на графическую, так и на вычислительную нагрузку.

Дополнительно в сцене применялись эффекты постобработки, включая базовые визуальные эффекты, типичные для современных игровых проектов. Их наличие и интенсивность учитывались как один из факторов, влияющих на итоговую производительность.

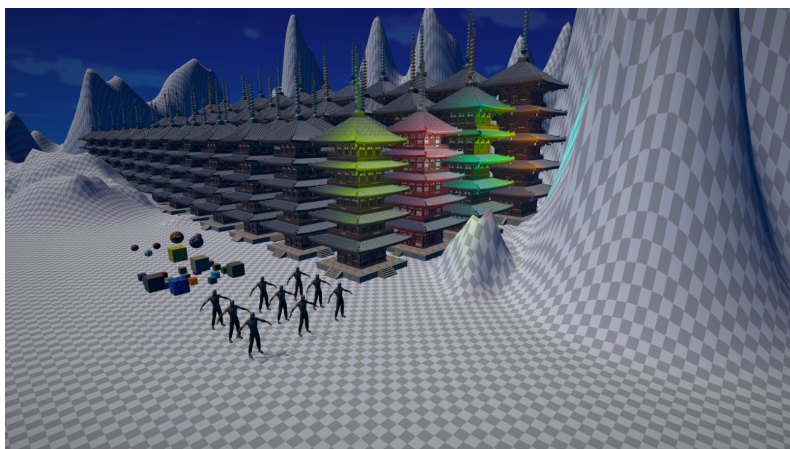


Рис. 1. Тестовая сцена

2.2. Режимы автоматизированного прогона сцены

В рамках исследования были реализованы два режима автоматизированного прогона сцены, отличающиеся способом задания параметров рендеринга, качества графики и контентной нагрузки.

В первом режиме использовался набор заранее определённых пресетов параметров, соответствующих дискретным уровням качества, предусмотренным в среде Unity3D. Каждый пресет представлял собой фиксированную комбинацию параметров рендеринга, освещения, постобработки и детализации геометрии. Данный режим обеспечивал воспроизводимость эксперимента и позволял выполнять прямое сравнение показателей производительности при идентичных конфигурациях сцены на различных аппаратных платформах.

Во втором режиме параметры сцены и рендеринга формировались стохастическим образом. Значения параметров выбирались псевдослучайно в пределах допустимых диапазонов, определённых настройками качества Unity3D и ограничениями экспериментальной сцены. Такой подход позволял получать разнообразные конфигурации сцены и формировать датасеты, отражающие широкий спектр возможных состояний среды выполнения.

Количество прогонов, длительность одного прогона и интервал сбора метрик задавались параметрами эксперимента и могли изменяться в зависимости от требований к точности и устойчивости измерений. Перед началом фиксации показателей для каждой конфигурации выполнялся этап прогрева, предназначенный для исключения влияния начальной загрузки ресурсов и инициализации сцены на результаты измерений.

Использование двух режимов автоматизированного прогона позволило, с одной стороны, анализировать детерминированные зависимости между параметрами сцены и показателями производительности, а с другой – формировать статистически более разнообразные датасеты, пригодные для последующего анализа и обучения моделей машинного обучения.

2.3. Система сбора и формирования данных производительности

Для автоматизированного сбора метрик производительности была разработана многоуровневая система мониторинга, включающая нативный и управляемый уровни. Низкоуровневый модуль обеспечивает получение информации о загрузке, частоте и температуре центрального и графического процессоров с использованием системных интерфейсов операционной системы и драйверов видеокарты. В случае недоступности аппаратных датчиков система корректно переходит в режим деградации без нарушения процесса эксперимента.

Взаимодействие с нативным модулем осуществляется из среды Unity через управляемый интерфейс. Основная логика мониторинга и управления сбором данных реализована в виде компонента сцены Unity. Для получения дополнительных показателей производительности используются встроенные средства профилирования, обеспечивающие доступ к графическим и временным метрикам в режиме выполнения приложения [1]. Сбор метрик осуществляется дискретно с фиксированным временным шагом в течение каждого прогона сцены. Для каждого временного

среза формируется запись, содержащая значения измеряемых показателей производительности и текущие параметры конфигурации сцены. По завершении прогона вычисляются агрегированные характеристики, включая средние значения и экстремумы метрик, используемые при формировании итогового датасета.

Сбор данных выполняется в рамках автоматизированного прогона набора конфигураций качества сцены. Каждая конфигурация задаётся дискретным набором параметров, влияющих на рендеринг, освещение, детализацию геометрии, постобработку и вычислительную нагрузку. Перед началом измерений параметры сцены программно устанавливаются, после чего выполняется этап прогрева.

В течение заданного интервала времени осуществляется периодический сбор временных срезов, включающих показатели частоты кадров, времени кадра, загрузки CPU и GPU, использования оперативной и видеопамати, а также графические метрики. По завершении прогона каждой конфигурации вычисляются агрегированные значения, после чего формируется единый датасет с метаданными устройства и параметрами эксперимента, сохраняемый в формате JSON (рис. 2).

На основе результатов автоматизированных прогонов формируется набор данных, который может быть представлен в виде множества наблюдений

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N,$$

где $\mathbf{x}_i$ – вектор признаков, описывающий параметры качества сцены и состояние среды выполнения в i -м прогоне, а y_i – агрегированная метрика производительности.

Вектор признаков $\mathbf{x}_i$ формируется следующим образом:

$$\mathbf{x}_i = (q_1, q_2, \dots, q_m, r_1, r_2, \dots, r_k),$$

где q_j – дискретные параметры качества рендеринга и контента (уровни теней, освещения, сглаживания, текстур, LOD, постобработки и плотности контента), а r_l – параметры состояния среды выполнения, включая фактический масштаб рендеринга, использование динамического разрешения и количество активных эффектов постобработки.

В качестве целевой переменной y_i в рамках настоящего исследования используется средняя частота кадров $\overline{FPS}_i$, вычисляемая по временным срезам, фиксируемым в процессе прогона сцены после завершения этапа прогрева.

Средняя частота кадров для i -й конфигурации определяется как

$$\overline{FPS}_i = \frac{1}{T} \sum_{t=1}^T FPS_{i,t},$$

где $FPS_{i,t}$ – значение частоты кадров, измеренное в момент времени t , а T – общее число временных срезов, зафиксированных в рамках одного прогона.

Таким образом, задача анализа производительности может быть сведена к исследованию зависимости

$$\overline{FPS} = f(\mathbf{x}),$$

где функция $f(\cdot)$ отражает влияние параметров качества сцены и состояния среды выполнения на итоговые показатели производительности. Сформированный датасет предназначен для выявления закономерностей данной зависимости и дальнейшего применения методов статистического анализа и машинного обучения.

3. Результаты экспериментальной апробации

В рамках экспериментальной апробации разработанная система сбора данных производительности была запущена на ограниченном наборе вычислительных устройств. Первичные прогоны были выполнены на авторском рабочем месте, после чего система была использована на устройствах нескольких независимых пользователей с различными аппаратными конфигурациями (рис. 3). Это позволило получить разнородные датасеты, отражающие влияние характеристик аппаратной платформы на показатели производительности. Для сопоставительной оценки вычислительных платформ и иллюстрации различий в производительности связок CPU+GPU были использованы открытые данные синтетических бенчмарков, опубликованные в специализированных онлайн-источниках [4–7].

Для повышения достоверности результатов анализ производительности выполнялся на основе агрегированных показателей, полученных по временным срезам в рамках каждого прогона. Использование средней частоты кадров позволяет снизить влияние кратковременных флуктуаций и обеспечить сопоставимость результатов между различными конфигурациями сцены и аппаратными платформами.

Следует отметить, что на ряде аппаратных конфигураций получение прямых показателей загрузки CPU и GPU оказалось недоступным в силу ограничений драйверов и системных интерфейсов. В этих случаях анализ производительности выполнялся на основе временных и графических метрик, таких как частота кадров и время кадра, что соответствует принятой практике оценки производительности игровых приложений.

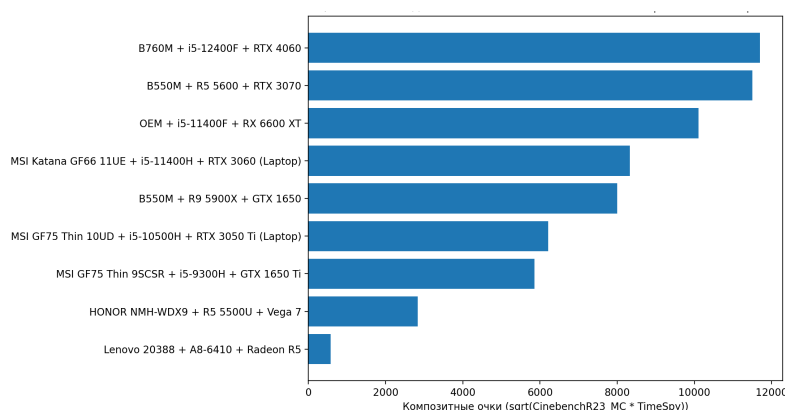


Рис. 3. Оценка производительности связок CPU+GPU (по открытым бенчмаркам)

На основе собранных данных был проведён первичный анализ, направленный на выявление зависимостей между параметрами вычислительных устройств и резуль-

```

"runId": "5e0e3b32dab74d1d9cd82c1e47e2c0cb",
"capturedAtUtc": "2025-12-29T06:44:40.5499406Z",
"datasetVersion": "2.0",
"device": {
  "deviceModel": "20388 (LENOVO)",
  "operatingSystem": "Windows 10 (10.0.19045) 64bit",
  "platform": "WindowsPlayer",
  "unityVersion": "2022.3.62f2",
  "applicationVersion": "0.1",
  "cpuName": "AMD A8-6410 APU with AMD Radeon R5 Graphics",
  "cpuCoreCount": 4,
  "cpuFrequencyMhz": 1996,
  "gpuName": "AMD Radeon(TM) R5 Graphics",
  "graphicsApi": "Direct3D11",
  "graphicsDeviceVersion": "Direct3D 11.0 [level 11.1]",
  "graphicsMemoryMb": 3566,
  "systemMemoryMb": 7132,
  "maxTextureSize": 16384,
  "supportsComputeShaders": true,
  "supportsInstancing": true,
  "supportsAsyncCompute": false,
  "qualityLevelIndex": 5,
  "qualityLevelName": "Ultra",
  "sceneName": "SampleScene",
  "screenResolution": "1366x768@60Hz",
  "screenDpi": 96.0,
  "graphicsMultiThreaded": false,
  "refreshRateHz": 60.0,
  "deviceFingerprint": "0d0de321360dcf15"
},
"totalConfigurations": 44,
"requestedConfigurations": 44,
"configurationDuration": 25.0,
"sampleInterval": 1.0,
"warmupDuration": 5.0,
"randomSeed": 1281278506,
"totalInvalidSamples": 1144,
"totalCappedSamples": 371,
"appliedConfigurationDuration": 25.0,
"appliedSampleInterval": 1.0,
"appliedWarmupDuration": 5.0,
"randomizeQualityPerConfiguration": true,
"restoreQualitySettingsOnFinish": true,
"configurations": [
  {
    "index": 0,
    "startedAtUtc": "2025-12-29T06:21:29.5479134Z",
    "quality": {
      "shadows": 2,
      "lighting": 2,
      "antiAliasing": 2,
      "textures": 2,
      "anisotropicFiltering": 2,

```

Рис. 2. Фрагмент JSON-файла с метаданными устройства и параметрами эксперимента

татами тестирования сцены при различных конфигурациях качества. Полученные датасеты послужили основой для построения первых графиков, иллюстрирующих характер взаимосвязей между аппаратными характеристиками и измеряемыми метриками производительности, а также подтвердили работоспособность и практическую применимость разработанной системы (рис. 4).

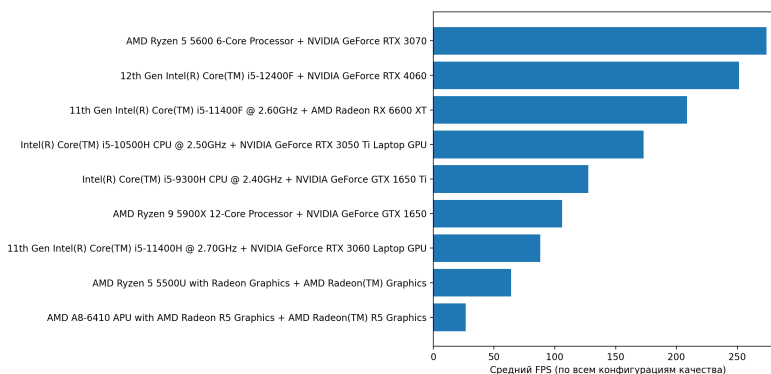


Рис. 4. Средняя производительность связок CPU+GPU по датасетам

Было зафиксировано устойчивое различие в средней частоте кадров между конфигурациями с идентичными параметрами сцены при изменении аппаратной платформы, что подтверждает чувствительность системы к характеристикам CPU и GPU.

Заключение

В рамках данной работы была разработана и апробирована система динамического профилирования игровых сцен в среде Unity3D, обеспечивающая автоматизированный сбор и агрегацию метрик производительности в режиме выполнения приложения. Предложенный подход позволяет выполнять многократный прогон сцены с систематическим изменением параметров рендеринга, контента и вычислительной нагрузки, формируя структурированный датасет, пригодный для последующего анализа и использования в задачах машинного обучения.

В ходе экспериментальной апробации было подтверждено, что система корректно функционирует на различных аппаратных конфигурациях и позволяет фиксировать зависимость показателей производительности от параметров сцены и характеристик вычислительных устройств. Полученные датасеты отражают влияние качества графики, сложности контента и аппаратной платформы на ключевые метрики, включая частоту кадров, загрузку CPU и GPU, а также использование памяти.

Практическая значимость работы заключается в возможности перехода от ручного профилирования отдельных конфигураций к систематическому анализу производительности, основанному на данных. Разработанная система может быть использована как инструмент поддержки принятия решений на этапе разработки игровых приложений, а также как основа для построения предиктивных моделей, оценивающих влияние параметров сцены на производительность.

В дальнейшем планируется расширение объёма собираемых данных, увеличение разнообразия тестовых сцен и аппаратных платформ, а также непосредственное применение полученных датасетов для обучения и валидации моделей машинного обучения, направленных на автоматизированную оптимизацию производительности игровых приложений.

Литература

1. Unity Technologies. Unity Manual: Profiler. URL: <https://docs.unity3d.com/Manual/Profiler.html> (дата обращения: 01.01.2026).
2. Unity Technologies. Unity Manual: Quality Settings. URL: <https://docs.unity3d.com/Manual/class-QualitySettings.html> (дата обращения: 02.01.2026).
3. Unity Technologies. Unity Manual: Graphics Performance and Optimization. URL: <https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html> (дата обращения: 03.01.2026).
4. NotebookCheck. AMD A8-6410 vs AMD Athlon 3050e: Benchmarks and Specifications. URL: https://www.notebookcheck.net/A8-6410-vs-Athlon-3050e_6219_12722_247596.0.html (дата обращения: 04.01.2026).
5. NotebookCheck. Intel Core i5-12400F Processor: Benchmarks and Specifications. URL: <https://www.notebookcheck.net/Intel-Core-i5-12400F-Processor-Benchmarks-and-Specs.825462.0.html> (дата обращения: 04.01.2026).
6. CPU Monkey. AMD Ryzen 5 5600 – Cinebench R23 Multi-Core Benchmark. URL: https://www.cpu-monkey.com/en/benchmark-amd-ryzen_5_5600-cinebench_r23_multi_core (дата обращения: 05.01.2026).
7. NotebookCheck. NVIDIA GeForce RTX 4060 GPU: Benchmarks and Specifications. URL: <https://www.notebookcheck.net/NVIDIA-GeForce-RTX-4060-GPU-Benchmarks-and-Specs.741899.0.html> (дата обращения: 05.01.2026).
8. Dickinson C. Unity 5 Game Optimization. Birmingham; Mumbai: Packt Publishing, 2015. 306 p.
9. Dickinson C. Unity 2017 Game Optimization. Birmingham; Mumbai: Packt Publishing Ltd., 2017. 312 p.

UNITY3D RUNTIME PROFILING WITH DATASET GENERATION FOR ML-BASED OPTIMIZATION

V.V. Firstov

Ph.D. Student, e-mail: ckontro@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The paper presents a system for automated runtime collection of performance data from Unity3D scenes. The solution executes multiple runs of a scene with dynamically changing rendering, content, and environment parameters, capturing key metrics such as FPS, CPU/GPU load, memory usage, and scene characteristics. The collected data are aggregated into a structured dataset for subsequent machine learning, enabling a data-driven alternative to manual profiling and laying the groundwork for predictive models that estimate performance impacts during development.

Keywords: performance optimization, game development, Unity3D, machine learning, dataset.

Дата поступления в редакцию: 05.01.2026

СРАВНЕНИЕ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ АНАЛИЗА СООТВЕТСТВУЮЩИХ ГРАФОВ

С.Н. Чуканов¹

д.т.н., профессор, ведущий научный сотрудник, e-mail: ch_sn@mail.ru

И.С. Чуканов²

магистрант, e-mail: chukanov022@gmail.com

С.В. Лейхтер³

старший преподаватель, e-mail: leykhter@mail.ru

¹Институт математики им. С.Л. Соболева СО РАН, Омский филиал, Омск, Россия

²Уральский федеральный университет имени первого Президента России Б.Н. Ельцина,
Екатеринбург, Россия

³Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. Рассматривается сравнение временных рядов совокупности концепций на основе формирования взвешенных ориентированных графов взаимных влияний концепций временных рядов друг на друга; при этом вершинам графа соответствуют концепции, а взвешенным дугам значения величин влияния одной концепции на другую. Структуру графа можно охарактеризовать с помощью спектра собственных значений матрицы Лапласа графа, связанного с уравнением теплопроводности. Для этого формируется матрица теплового ядра; производная матрицы теплового ядра по времени определяется матрицей Лапласа. Коэффициенты разложения теплового содержания теплового ядра в ряд Тейлора можно использовать для представления структуры графа. В работе представлены соотношения для определения расстояния Махаланобиса между векторами признаков – инвариантами теплового ядра, которые могут быть использованы для распознавания образов. Актуальность проблемы – повышение уникальности идентификации при распознавании временных рядов за счет учета как спектра собственных значений матрицы Лапласа графа, так и собственных векторов. Цель исследования – выяснить возможность использования коэффициентов степенного ряда разложения теплового содержания построенного графа в качестве векторов признаков – характеристик свойств графа. Новизна представленных в работе методов заключается в том, что, в отличие от метода спектрального вложения, в работе исследуется возможность сравнения графов (изображений) на основе расстояния Махаланобиса между инвариантами, построенными по коэффициентам разложения теплового содержания. Предлагается алгоритм минимизации расстояния Махаланобиса между вершинами графа, основанный на обучении матрицы Махаланобиса. На основе сравнения графов, соответствующих временным рядам совокупности концепций, можно сравнивать временные ряды.

Ключевые слова: тепловое ядро, расстояние Махаланобиса, временные ряды.

Введение

В работе рассматривается сравнение временных рядов совокупности концепций на основе формирования взвешенных ориентированных графов взаимных влияний концепций временных рядов друг на друга; при этом вершинам графа соответствуют концепции, а взвешенным дугам значения величин влияния одной концепции на другую. Структуру графа можно охарактеризовать с помощью спектра собственных значений матрицы Лапласа графа [1]. Спектр матрицы Лапласа связан с уравнением теплопроводности [2], которое может быть использовано для определения потока информации через сеть графа. Производная теплового ядра по времени определяется матрицей Лапласа. Тепловое содержание (сумма множества элементов теплового ядра) может быть разложено в степенной ряд по времени [4]. Так как решение уравнения теплопроводности является функцией времени, то в [3] рассматривается алгоритм определения матрицы теплового ядра для конкретного момента времени; выбор конкретного момента времени не указывается. В настоящей работе рассматривается определение коэффициентов разложения теплового содержания (сумма элементов матрицы теплового ядра по узлам графа) при разложении теплового содержания в степенной ряд по времени. Коэффициенты разложения теплового содержания теплового ядра в ряд Тейлора можно использовать для представления структуры графа. Для сравнения графов предлагается находить расстояние Махаланобиса [5, 6] между векторами признаков – коэффициентами разложения теплового содержания. Определение расстояния Махаланобиса между векторами признаков и элементарными симметрическими многочленами лапласианов графов может быть использовано для распознавания формы объектов. Цель работы – выяснить возможность использования коэффициентов степенного ряда разложения теплового содержания построенного графа в качестве векторов признаков – характеристик свойств графа. Новизна представленных в работе методов заключается в том, что, в отличие от метода спектрального вложения, в работе исследуется возможность сравнения графов (изображений) на основе расстояния Махаланобиса между инвариантами, построенными по коэффициентам разложения теплового содержания. Предлагается алгоритм минимизации расстояния Махаланобиса между вершинами графа, основанный на обучении матрицы Махаланобиса. На основе сравнения графов, соответствующих временным рядам совокупности концепций, можно сравнивать временные ряды. Применение метода сравнения временных рядов основе построения и сравнения графов позволит идентифицировать штатное и нештатное функционирование динамических объектов, систем управления, приборного состава и т.д.

1. Тепловое ядро на графах

Пусть $G = \{V, E, W\}$ – неориентированный взвешенный граф, где $V = \{v_1, \dots, v_n\}$ – множество узлов графа, представленных в евклидовых координатах. $E = \{e_1, \dots, e_m\}$ – множество неориентированных рёбер, $W = [w_{ij}]$; $i, j \in \{1, \dots, n\}$ – квадратная взвешенная матрица смежности графа. Матрица Ла-

матрицы L графа может быть определена как:

$$L = D - W, \quad (1)$$

где $D \in \mathbb{R}^{n \times n}$ – диагональная матрица с элементами $D_i = \sum_{j=1}^n w_{ij}$. Вложение узлов графа в евклидово пространство может быть получено с помощью собственного спектра матрицы Лапласа графа. Пусть $L = U \Lambda U^T$ – это спектральное разложение матрицы L , где $U = [u_1, \dots, u_n]$ – собственные векторы, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ – собственные значения матрицы Лапласа L . Уравнение теплопроводности, связанное с матрицей Лапласа, может быть записано как:

$$\partial h_t / \partial t = -L h_t, \quad (2)$$

где h_t – тепловое ядро, соответствующее графу. Решение уравнения теплопроводности имеет вид: $h_t = \exp(-tL) \in \mathbb{R}^{n \times n}$. Определим тепловое ядро на графе на основе спектра Лапласа [3]:

$$h_t(i, j) = \Phi \exp(-\Lambda t) \Phi^T = \sum_{l=1}^n \exp(-\lambda_l t) u_{li} u_{lj}^T. \quad (3)$$

Тепловое содержание – это сумма элементов матрицы теплового ядра по узлам графа:

$$Q(t) = \sum_i \sum_j h_t(i, j) = \sum_i \sum_j \sum_{k=1}^n e^{-\lambda_k t} u_{ik} u_{jk}. \quad (4)$$

Тепловое содержание можно разложить в степенной ряд по времени [4]:

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m. \quad (5)$$

После разложения экспоненты $e^{-\lambda_k t}$ в ряд Тейлора коэффициенты q_m разложения $Q(t)$ можно представить следующим образом:

$$q_m = \sum_{i,k} \frac{(-\lambda_k)^m}{m!} \left(\sum_i \sum_j u_{ik} u_{jk} \right) = \sum_{k=1}^n \frac{(-\lambda_k)^m}{m!} \left(\sum_i u_{ik} \right)^2. \quad (6)$$

Коэффициенты q_m зависят от собственных значений и собственных векторов; они более уникальны, чем инварианты, основанные только на собственных значениях. Пусть q_1 – вектор коэффициентов разложения $Q(t)$, построенного по лапласиану для графа G^1 ; q_2 – вектор коэффициентов, построенного по лапласиану для графа G^2 . Расстояние Махаланобиса (взвешенное евклидово расстояние) между векторами коэффициентов q_1 и q_2 , позволяющее различать графы G^1 и G^2 , можно определить по формуле [5, 6]:

$$d(q_1, q_2)^2 = \text{trace} \left((q_1 - q_2) W_q (q_1 - q_2)^T \right), \quad (7)$$

где $W_q = \text{diag} \begin{pmatrix} a_1 & a_2 & a_3 & \dots \end{pmatrix}$ – диагональная матрица Махаланобиса положительных весов. Поставим задачу обучения матрицы W_q , основанную на минимизации расстояния $d(q_1, q_2)^2$ путём выбора матрицы

$$W_q : \min_A \text{trace} \left((q_1 - q_2) W_q (q_1 - q_2)^T \right), \quad (8)$$

при ограничении: $\sum_i (W_q)_i = 1$.

2. Сравнение временных рядов

Нечеткие когнитивные карты состоят из концепций $C_i, i = 1, \dots, n$. Каждый узел (концепция) содержит значение $A_i \in [0, 1], i = 1, \dots, n$. Концепции соединены взвешенными ориентированными дугами, которые показывают отношения между концептами. Дуга между концепциями A_i и A_j имеет вес w_{ij} . Направление указывает, является ли концепция причиной или следствием. На каждом шаге значение концепции A_i находится под влиянием значений других концепций:

$$A_i(k+1) = f \left(A_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^n w_{ji} A_j(k) \right), i = 1, \dots, n, \quad (9)$$

где w_{ij} – вес дуги, соединяющей концепцию причины C_j с концепцией следствия C_i ; $f(x)$ – сигмоидальная функция: $f(x) = (1 + \exp(-\lambda(x - \varsigma)))^{-1}$, где λ – значение крутизны сигмоидальной функции, ς – значение, при котором $f(\sigma) = 0.5$. Процесс обучения используется для определения матрицы весов $W = [w_{ij}]$. Доступные для обучения данные поступают в виде последовательности векторов $A_c(k), k = 1, 2, \dots, N$. Функция критерия, по экстремуму (минимуму) которой определяются требуемые значения элементов матрицы W , определяется как сумма расстояний между прогнозируемыми векторами активации узлов $A(k+1)$, которые вычисляются на основе $A(k)$, доступных в k -й момент времени, и векторами $\tilde{A}(k+1)$ известных уровней активации узлов $(k+1)$ -й момент времени:

$$V = \min_{w_{ij}} \left(\sum_{k=1}^N \sum_{i=1}^n \left\| \tilde{A}_i(k+1) - f_i \left(\sum_{j=1}^n w_{ij} A_j(k) \right) \right\|^2 \right). \quad (10)$$

Рассмотрим пример оценивания весов взаимного влияния концепций экологии города [7, 8] по информации о временных рядах этих концепций. Введем следующие обозначения концепций: A_1 – население города; A_2 – миграция в город; A_3 – модернизация; A_4 – мусор на единицу площади; A_5 – санитарные условия; A_6 – количество заболеваний; A_7 – количество бактерий на единицу площади. В таблице 1 заданы исходные временные ряды концепций $A_1 - A_7$ при компонентах матрицы взаимных

Таблица 1. Исходный временной ряд

t	A1	A2	A3	A4	A5	A6	A7
0	0.300	0.350	0.400	0.450	0.500	0.550	0.600
1	0.166	0.634	0.579	0.760	0.872	0.597	0.568
2	0.098	0.934	0.715	0.890	0.988	0.236	0.460
3	0.132	0.991	0.774	0.903	0.997	0.019	0.345
4	0.208	0.994	0.837	0.941	0.997	0.004	0.232
5	0.283	0.996	0.907	0.963	0.998	0.002	0.169
6	0.357	0.997	0.943	0.971	0.999	0.002	0.140

Таблица 2. Исходные значения весов $W_0 = [w_{ij}]$

	A1	A2	A3	A4	A5	A6	A7
A1	0	0.1	0	0	0	-0.3	0
A2	0	0	0.7	0	0	0	0
A3	0.6	0	0	0	0	0	0
A4	0.9	0	0	0	0	0	0
A5	0	0	0.9	0	0	0	0
A6	0	0	0	0	-0.9	0	0.8
A7	0	0	0	0.9	-0.9	0	0

влияний, заданных в таблице 2. Определим веса w_{ij} взаимных влияний концепций по информации таблицы 1.

Определим матрицу Лапласа из соотношения (1):

$$L_0 = \begin{pmatrix} -0.2 & -0.3 & -0.2 & -0.2 & 0 & 0.1 & 0 \\ 0.7 & 0.7 & 0 & 0.7 & 0.7 & 0.7 & 1 \\ 0 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 1 \\ 0 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 1 \\ 0.9 & 0.9 & 0 & 0.9 & 0.9 & 0.9 & 1 \\ -0.1 & -0.1 & -0.1 & -0.1 & 0.8 & 0 & -1 \\ 0 & 0 & 0 & -0.9 & 0.9 & 0 & 0 \end{pmatrix}.$$

Собственные значения $e_0 = \text{eig}(L_0) =$

$$= \begin{pmatrix} 2.7123 & 0.7474 + 0.5012i & 0.7474 - 0.5012i & -0.5862 + 0.4398i & -0.5862 - 0.4398i & -0.1346 & 0 \end{pmatrix}.$$

Определим коэффициенты q разложения $Q(t)$ для лапласиана L_0 по формуле (7):

$$q_0 = \begin{pmatrix} -9.4853 & 0.0196 + 0.0477i & 0.0031 + 0.0152i & -0.0012 - 0.0008i & -0.0002 & 0 & 0 \end{pmatrix}.$$

Таблица 3. Исходный временной ряд

t	A1	A2	A3	A4	A5	A6	A7
0	0.300	0.350	0.400	0.450	0.500	0.550	0.600
1	0.153	0.656	0.592	0.750	0.853	0.614	0.589
2	0.085	0.947	0.735	0.877	0.989	0.294	0.527
3	0.116	0.993	0.813	0.901	0.997	0.034	0.443
4	0.187	0.994	0.878	0.933	0.997	0.007	0.364
5	0.258	0.995	0.921	0.954	0.998	0.004	0.306
6	0.331	0.997	0.947	0.964	0.999	0.003	0.269

Таблица 4. Исходные значения весов $W_0 = [w_{ij}]$

	A1	A2	A3	A4	A5	A6	A7
A1	0	0.097	0	0	0	-0.312	0
A2	0	0	0.681	0	0	0	0
A3	0.580	0	0	0	0	0	0
A4	0.885	0	0	0	0	0	0
A5	0	0	0.889	0	0	0	0
A6	0	0	0	0	-0.901	0	0.819
A7	0	0	0	0.925	-0.892	0	0

Определим матрицу Лапласа из соотношения (1):

$$L_1 = \begin{pmatrix} -0.173 & -0.28 & -0.173 & -0.173 & -0.173 & 0.1071 & -0.173 \\ 0.704 & 0.704 & 0 & 0.704 & 0.7042 & 0.7042 & 0.704 \\ 0 & 0.6 & 0.6 & 0.6 & 0.5998 & 0.5998 & 0.6 \\ 0 & 0.894 & 0.894 & 0.894 & 0.8941 & 0.8941 & 0.894 \\ 0.913 & 0.913 & 0 & 0.913 & 0.9129 & 0.9129 & 0.913 \\ -0.055 & -0.055 & -0.055 & -0.055 & 0.8212 & -0.055 & -0.876 \\ -0.012 & -0.012 & -0.012 & -0.918 & 0.9055 & -0.012 & -0.012 \end{pmatrix}.$$

Собственные значения $e_1 = \text{eig}(L_1) =$

$$= \begin{pmatrix} 2.6911 & -0.5909 + 0.4714i & -0.5909 - 0.4714i & 0.6422 + 0.4553i \\ 0.6422 - 0.4553i & 0.0064 & 0 & 0 \end{pmatrix}.$$

Определим коэффициенты q разложения $Q(t)$ для лапласиана L_1 по формуле (7):

$$q_1 = \begin{pmatrix} -9.6948 & 0.0060 - 0.0224i & -0.0022 + 0.0052i & -0.0005 + 0.0004i & 0.0001 & 0 & 0 \end{pmatrix},$$

тогда

$$q_1 - q_0 =$$

$$= \begin{pmatrix} -0.2091 & -0.0135 - 0.0701i & -0.0052 - 0.0099i & 0.0007 + 0.0012i & 0.0003 & 0 & 0 \end{pmatrix}.$$

Решая задачу минимизации (9), найдём матрицу Махаланобиса: $W = \text{diag}(0.0036 \ 0.0081 \ 0.0324 \ 0.1089 \ 0.1156 \ 0 \ 0)$ и расстояние Махаланобиса между векторами коэффициентов q_1 и q_0 : $WM_1 = (q_1 - q_0) \cdot W \cdot (q_1 - q_0)^T = 0.000203$.

Таблица 5. Исходный временной ряд

t	A1	A2	A3	A4	A5	A6	A7
0	0.300	0.350	0.400	0.450	0.500	0.550	0.600
1	0.170	0.646	0.636	0.770	0.842	0.624	0.488
2	0.111	0.961	0.761	0.890	0.988	0.251	0.281
3	0.169	0.993	0.843	0.907	0.997	0.011	0.106
4	0.288	0.996	0.894	0.948	0.997	0.002	0.046
5	0.430	0.997	0.951	0.968	0.998	0.001	0.041
6	0.601	0.998	0.969	0.983	0.998	0.001	0.044

Таблица 6. Исходные значения весов $W_0 = [w_{ij}]$

	A1	A2	A3	A4	A5	A6	A7
A1	0	0.151	0	0	0	-0.308	0
A2	0	0	0.727	0	0	0	0
A3	0.619	0	0	0	0	0	0
A4	0.841	0	0	0	0	0	0
A5	0	0	0.801	0	0	0	0
A6	0	0	0	0	-0.890	0	0.827
A7	0	0	0	0.860	-0.991	0	0

Определим матрицу Лапласа из соотношения (1):

$$L_2 = \begin{pmatrix} -0.157 & -0.308 & -0.157 & -0.157 & -0.157 & 0.1506 & -0.157 \\ 0.727 & 0.727 & 0 & 0.727 & 0.7266 & 0.7266 & 0.727 \\ 0 & 0.619 & 0.619 & 0.619 & 0.619 & 0.619 & 0.619 \\ 0 & 0.841 & 0.841 & 0.841 & 0.841 & 0.841 & 0.841 \\ 0.801 & 0.801 & 0 & 0.801 & 0.801 & 0.801 & 0.801 \\ -0.063 & -0.063 & -0.063 & -0.063 & 0.8268 & -0.063 & -0.89 \\ -0.131 & -0.131 & -0.131 & -0.991 & 0.8595 & -0.131 & -0.131 \end{pmatrix}.$$

Собственные значения $e_2 = \text{eig}(L_2) =$

$$= \begin{pmatrix} 2.7151 & -0.5839 + 0.4793i & -0.5839 - 0.4793i & 0.6149 + 0.4594i \\ & 0.6149 - 0.4594i & 0.0229 & 0 \end{pmatrix}.$$

Определим коэффициенты q разложения $Q(t)$ для лапласиана L_2 по формуле (7):

$$q_2 = \begin{pmatrix} -8.1360 & 0.0063 - 0.0282i & -0.0032 + 0.0065i & -0.0006 + 0.0005i & 0.0001 & 0 & 0 \end{pmatrix},$$

тогда

$$q_2 - q_0 = \begin{pmatrix} 1.3493 & -0.0133 - 0.0759i & -0.0063 - 0.0086i & 0.0005 + 0.0012i & 0.0003 & 0 & 0 \end{pmatrix}.$$

При матрице Махаланобиса: $W = \text{diag}(0.0036 \ 0.0081 \ 0.0324 \ 0.1089 \ 0.1156 \ 0 \ 0)$ найдём расстояние Махаланобиса между векторами коэффициентов q_2 и q_0 : $WM_2 = (q_2 - q_0) \cdot W \cdot (q_2 - q_0)^T = 0.0066$. Так как $WM_2 / WM_1 \approx 32.5$, расстояние Махаланобиса WM_2 больше расстояния Махаланобиса WM_1 .

Заключение

В статье рассмотрена возможность сравнения временных рядов совокупности концепций. Для сравнения временных рядов предлагается формировать взвешенные ориентированные графы взаимных влияний концепций временных рядов друг на друга: вершинам графа соответствуют концепции, а взвешенным дугам значения величин влияния одной концепции на другую. Рассмотрено использование инвариантов, определяемых из теплового ядра уравнения теплопередачи, построенного на основе матрицы Лапласа графов, для характеристики графов. Применение метода сравнения временных рядов основе построения и сравнения графов позволит идентифицировать штатное и нештатное функционирование динамических объектов, систем управления, приборного состава и т.д. Предложен алгоритм минимизации расстояния Махаланобиса между вершинами графа, основанный на обучении матрицы Махаланобиса.

Благодарности

Работа выполнена в рамках федерального проекта «Экономика данных и цифровая трансформация государства»: «Кадры для цифровой трансформации», грант на обеспечение обучения студентов по образовательным программам высшего образования для топ-специалистов в сфере информационных технологий на базе АНО «Аналитический центр при Правительстве Российской Федерации» ИГК по договору ИТ: 000000Ц830325P2M0002.

Литература

1. Chung F.R.K. Spectral graph theory. American Mathematical Soc., 1997. V. 92.
2. Gilkey P.B. Invariance theory: the heat equation and the Atiyah-Singer index theorem. CRC press, 2018.
3. Xiao B., Hancock E.R., Wilson R.C. Graph characteristics from the heat kernel trace // Pattern Recognition. 2009. V. 42, No. 11. P. 2589–2606.
4. McDonald P., Meyers R. Isospectral polygons, planar graphs and heat content // Proceedings of the American mathematical society. 2003. V. 131, No. 11. P. 3589–3599.
5. Deza M.M., Deza E. Encyclopedia of Distances. Springer: Berlin-Heidelberg, 2016.
6. Mahalanobis P.C. On the generalised distance in statistics // Sankhya, A 80. 1936. P. 1–7.
7. Lu W. et al. The modeling and prediction of time series based on synergy of high-order fuzzy cognitive map and fuzzy c-means clustering // Knowledge-Based Systems. 2014. V. 70. P. 242–255.
8. Hagiwara M. Extended fuzzy cognitive maps // IEEE Transactions on Electronics, Information and Systems. 1994. V. 114, No. 3. P. 367–372.

COMPARISON OF TIME SERIES BASED ON THE ANALYSIS OF CORRESPONDING GRAPHS

S.N. Chukanov¹

Dr.Sc. (Techn.), Professor, Senior Scientist Researcher, e-mail: ch_sn@mail.ru

I.S. Chukanov²

master's student, e-mail: chukanov022@gmail.com

S.V. Leykhter³

Senior Lecturer, e-mail: leykhter@mail.ru

¹Sobolev Institute of Mathematics, Omsk branch, Russia

²Ural Federal University named after the first President of Russia B.N. Yeltsin, Ekaterinburg, Russia

³Omsk State University F.M. Dostoevsky

Abstract. This paper examines the comparison of time series of a set of concepts based on the formation of weighted directed graphs of the mutual influences of time series concepts on each other. The graph vertices correspond to the concepts, and the weighted arcs correspond to the values of the influence of one concept on another. The graph structure can be

characterized using the spectrum of the eigenvalues of the graph's Laplace matrix, which is related to the heat equation. For this purpose, a thermal kernel matrix is formed; the time derivative of the thermal kernel matrix is determined by the Laplace matrix. The coefficients of the Taylor series expansion of the thermal content of the thermal kernel can be used to represent the graph structure. The paper presents relations for determining the Mahalanobis distance between feature vectors thermal kernel invariants – which can be used for pattern recognition. The relevance of the problem is to increase the uniqueness of identification in time series recognition by taking into account both the spectrum of the eigenvalues of the graph's Laplace matrix and its eigenvectors. The aim of this study is to explore the feasibility of using the power series coefficients of the heat content expansion of the constructed graph as feature vectors—characteristics of the graph's properties. The novelty of the methods presented in this paper lies in the fact that, unlike the spectral embedding method, it explores the possibility of comparing graphs (images) based on the Mahalanobis distance between invariants constructed from the heat content expansion coefficients. An algorithm for minimizing the Mahalanobis distance between graph vertices is proposed, based on learning the Mahalanobis matrix. By comparing graphs corresponding to time series of a set of concepts, it is possible to compare time series.

Keywords: heat kernel on a graph, Mahalanobis distance, time series comparison.

Дата поступления в редакцию: 05.01.2026

Научный журнал

Математические структуры и моделирование

№ 1 (77)

Главный редактор

А.К. Гуц

Зам. гл. ред., выпускающий редактор
Д.Н. Лавров

Зам. гл. ред., технический редактор
Н.Ф. Богаченко

Авторская редакция

Адрес научной редакции

644077, Омская обл., г. Омск, пр-т Мира, д. 55а,
Омский государственный университет

E-mail: dmtr1@ya.ru

Электронная версия журнала:

<http://msm.omsu.ru>



Подписано в печать 04.05.2026. Формат 60 × 84 1/8.

Усл. печ. л. 15,22. Тираж 25 экз. Заказ № 142

Отпечатано на полиграфической базе издательства ОмГУ им. Ф.М. Достоевского
644077, Омская обл., г. Омск, пр-т Мира, д. 55а

ISSN 2222-8772



9 772222 877005



26077 >