

Министерство науки и высшего образования РФ
Смоленский государственный университет

Системы компьютерной математики и их приложения

Выпуск 27
Часть 1

Смоленск
Издательство СмолГУ
2026

УДК 621.396.218
ББК 32.97
С 409

*Печатается по решению
редакционно-издательского
совета СмолГУ*

Редакционная коллегия

*К.М. Расулов, д-р физ.-мат. наук, проф. (ответственный редактор)
Н.М. Тимофеева, канд. пед. наук, доц. (ответственный секретарь)
В.В. Борисов, д-р техн. наук, проф. (Смоленск, Россия)
Ю.М. Вувуникян, д-р физ.-мат. наук, проф. (Гродно, Республика Беларусь)
Л.Г. Гагарина, д-р техн. наук, проф. (Зеленоград, Россия)
С.А. Гомонов, канд. физ.-мат. наук, доц. (Смоленск, Россия)
А.П. Карпенко, д-р физ.-мат. наук, проф. (Москва, Россия)
А.Б. Костин, д-р физ.-мат. наук, проф. (Москва, Россия)
В.И. Мунерман, канд. техн. наук, доц. (Смоленск, Россия)
Г.Е. Сенькина, д-р пед. наук, проф. (Смоленск, Россия)
И.В. Тихонов, д-р физ.-мат. наук, проф. (Москва, Россия)
В.Б. Шерстюков, д-р физ.-мат. наук, проф. (Москва, Россия)*

С 409 **Системы компьютерной математики и их приложения:**
межвузовский сборник научных трудов. Смоленск: Изд-во
СмолГУ, 2026. Вып. 27. Часть 1. 319 с.
ISBN 978-5-88018-445-3, *продолжающееся издание*

В настоящем выпуске сборника публикуются расширенные тексты научных докладов, представленных на XXVII Международной научной конференции «Системы компьютерной математики и их приложения», проходившей 22–23 мая 2026 года в г. Смоленске на базе физико-математического факультета Смоленского государственного университета. В работе конференции приняли участие научные сотрудники и преподаватели вузов ряда стран СНГ.

В материалах сборника рассматриваются вопросы применения систем компьютерной математики и их приложений в различных областях науки и техники, в математическом, техническом и гуманитарном образовании.

Сборник рекомендуется научным работникам, преподавателям вузов, аспирантам и студентам старших курсов университетов.

УДК 621.396.218
ББК 32.97

ISBN 978-5-88018-445-3,
продолжающееся издание

© Авторы, 2026
© Издательство СмолГУ, 2026

РАЗДЕЛ 1

Системы компьютерной математики

А.Р. Андреев, А.А. Бондарь, О.В. Тихонова
Российский технологический университет МИРЭА

УДК 621.391

РАЗРАБОТКА RAPTOR-КОДЕРА ДЛЯ РЕАЛИЗАЦИИ НА ПЛИС

Ключевые слова: *Raptor-код, фонтанные коды, помехоустойчивое кодирование, LDPC-предкодер, LT-кодер, ПЛИС.*

В данной статье рассматривается реализация Raptor-кодера, предназначенного для помехоустойчивой передачи данных в каналах связи с потерями пакетов. Кодер реализован на языке Verilog в среде Xilinx Vivado. Он состоит из LDPC-предкодера и LT-кодера. Приводится описание Raptor-кодирования, описывается архитектура разработанного кодера и рассматриваются особенности реализации алгоритма.

1. Методы помехоустойчивого кодирования в системах передачи данных. Современные системы передачи данных часто функционируют в условиях нестабильных каналов связи. Потери пакетов характерны для беспроводных сетей, сетей доставки мультимедийного контента и распределённых вычислительных систем. Для повышения надёжности передачи используются методы помехоустойчивого кодирования.

Классические методы кодирования, такие как коды Хэмминга, ВСН-коды и коды Рида – Соломона, требуют заранее заданного объёма избыточности. В условиях непредсказуемых потерь пакетов это может приводить либо к недостаточной защите данных, либо к избыточному увеличению объёма передаваемой информации.

Одним из перспективных направлений в теории помехоустойчивого кодирования являются фонтанные коды. Основная особенность таких кодов – возможность генерации потенциально неограниченного числа кодовых символов на основе фиксированного набора исходных данных. Получатель может восстановить исходное сообщение после получения любого достаточного количества кодовых символов.

Первым практически реализуемым фонтанным кодом стал LT-код, предложенный М. Луби. Однако декодирование LT-кодов требует достаточно большого числа операций. Для повышения эффективности был разработан Raptor-код, который сочетает предкодирование исходных данных и LT-кодирование промежуточных символов. Предкодер формирует дополнительные проверочные символы, что значительно улучшает свойства декодирования. LT-кодер, в свою очередь, генерирует последовательность кодовых символов, каждый из которых является линейной комбинацией нескольких промежуточных символов.

2. Принцип работы Raptor-кодера. Разработанный кодер реализует каскадную архитектуру фонтанного кода типа Raptor. Кодер состоит из двух основных функциональных блоков: предкодера на основе LDPC-кода и LT-кодера, формирующего выходные кодовые символы. Оба блока реализованы в виде отдельных IP-ядер в среде проектирования Xilinx Vivado и объединены верхнеуровневым модулем raptor_encoder_top. Такая архитектура соответствует классической структуре Raptor-кодов, в которой сначала выполняется линейное предкодирование исходных символов, после чего применяется фонтанный LT-код.

На первом этапе кодирования исходные данные поступают на вход предкодера. В разработанной реализации входной блок исходных данных состоит из K символов фиксированной длины одного символа 16 бит. В рассматриваемой конфигурации используются параметры:

$K = 4$ - число исходных символов;

$S = 2$ - число LDPC-символов;

$N = 3$ - число дополнительных half-символов.

Исходные данные подаются на вход модуля в виде объединённой шины. Далее выполняется их разделение на отдельные символы s_0, s_1, s_2, s_3 . После этого формируются проверочные символы LDPC-кода. В реализованной модели они вычисляются как побитовые операции XOR между исходными символами:

$$\begin{aligned} p_0 &= s_0 \oplus s_1 \\ p_1 &= s_2 \oplus s_3 \end{aligned} \tag{1}$$

Такие операции реализуют простые линейные проверочные соотношения, характерные для кодов с низкой плотностью проверочной матрицы. Полученные проверочные символы обеспечивают дополнительную избыточность, повышающую вероятность восстановления исходных данных при потере части пакетов.

После формирования LDPC-символов вычисляется набор дополнительных half-символов. Они представляют собой линейные комбинации исходных и проверочных символов:

$$\begin{aligned}
h_0 &= s_0 \oplus s_2 \oplus p_0 \\
h_1 &= s_1 \oplus s_3 \oplus p_1 \\
h_2 &= s_0 \oplus s_3 \oplus p_0 \oplus p_1
\end{aligned}
\tag{2}$$

Таким образом формируется полный набор промежуточных символов:

$$L = K + S + H. \tag{3}$$

Эти символы объединяются в единый вектор `intermediate_data` в следующем порядке: исходные символы, LDPC-символы, half-символы. После поступления сигнала вычисленные промежуточные символы записываются в регистр и становятся доступными для следующего этапа кодирования.

На втором этапе кодирования используется LT-кодер, который формирует последовательность выходных кодовых символов на основе промежуточных символов. Каждый новый выходной символ генерируется по запросу сигнала.

При LT-кодировании происходит случайный выбор набора промежуточных символов, которые объединяются для формирования выходного символа. Для каждого символа определяется идентификатор кодового символа ESI. На основе значения ESI генерируется псевдослучайное число, значение которого используется для определения параметров формирования выходного символа. Сначала по таблице распределения степеней определяется степень символа d , которая показывает количество промежуточных символов, участвующих в формировании выходного символа. Таблица реализует упрощённое распределение Robust Soliton.

Далее вычисляются параметры a – начальный индекс и b – шаг выбора символов, задающие последовательность индексов промежуточных символов: a, b – шаг выбора символов. Индексы промежуточных символов определяются выражением

$$index_i = (a + i \cdot b) \bmod L, \tag{4}$$

где $i = 0 \dots d - 1$.

Каждый выбранный промежуточный символ объединяется с предыдущими при помощи операции XOR. В результате формируется выходной кодовый символ

$$c = x_{index_0} \oplus x_{index_1} \oplus \dots \oplus x_{index_{d-1}}, \tag{5}$$

где x_j – промежуточные символы.

После вычисления символ передаётся на выход.

Тор-модуль выполняет координацию работы обоих IP ядер. Сначала по сигналу запускается процесс предкодирования и

формируется вектор промежуточных символов. После этого по сигналу LT-кодер многократно генерирует выходные символы. Такой принцип работы соответствует основной идее фонтанных кодов: кодер способен генерировать потенциально неограниченное число кодовых символов из одного блока исходных данных, что позволяет отправлять дополнительные символы до тех пор, пока декодер не сможет успешно восстановить исходное сообщение.

Для проверки работоспособности разработанного Raptor-кодера была реализована модель декодера LT-кода на языке Verilog. Декодер предназначен для восстановления промежуточных символов, сформированных предкодером, на основе полученных LT-символов. После восстановления всех промежуточных символов исходные данные могут быть получены напрямую, поскольку первые K элементов вектора промежуточных символов соответствуют исходным данным.

Декодер реализует итеративный алгоритм решения системы линейных уравнений. Каждый принятый LT-символ можно рассматривать как уравнение вида

$$c = x_{i_1} \oplus x_{i_2} \oplus \dots \oplus x_{i_d}, \quad (6)$$

где x_i – промежуточные символы, а d – степень LT-символа.

Задача декодера заключается в восстановлении значений всех промежуточных символов x_i . В реализованной архитектуре используется алгоритм итеративного исключения. Принцип работы алгоритма заключается в последовательном нахождении уравнений, содержащих только один неизвестный символ. Если такое уравнение найдено, соответствующий промежуточный символ может быть непосредственно вычислен. После этого найденное значение подставляется во все остальные уравнения, уменьшая их степень. Процесс повторяется до тех пор, пока не будут восстановлены все промежуточные символы или пока не станет невозможным дальнейшее уменьшение степени уравнений.

На вход LT-декодера поступают принятые кодовые символы и их идентификаторы ESI. Значение ESI используется для восстановления структуры связей между символами. Декодер применяет те же функции генерации, что и кодер: псевдослучайный генератор, таблицу распределения степеней и процедуру формирования параметров. Благодаря этому декодер может определить, какие именно промежуточные символы участвовали в формировании каждого LT-символа.

Используя значение ESI, декодер вычисляет параметры d , a , b , после чего определяет набор индексов промежуточных символов по выражению (4). Таким образом для каждого принятого символа

формируется уравнение вида (5). Далее выполняется процедура итеративного исключения неизвестных символов.

В разработанной реализации декодер содержит буфер для хранения принятых LT-символов и массив для хранения восстановленных промежуточных символов. Для каждого уравнения выполняется проверка количества уже известных промежуточных символов. Если в уравнении остается только один неизвестный символ, его значение может быть вычислено как XOR всех известных символов и принятого LT-символа. После определения нового символа соответствующий флаг вектора декодированных символов устанавливается в единицу, а процесс декодирования продолжается.

Декодирование завершается, когда восстановлены все L промежуточных символов. После завершения процесса декодирования первые K элементов массива промежуточных символов содержат исходные данные S_0, S_1, S_2, S_3 .

LT-декодер реализован в виде отдельного IP ядра, совместимого с ранее созданным кодером. Декодер принимает поток кодовых символов и их идентификаторы, восстанавливает промежуточные символы и формирует сигнал завершения декодирования после успешного восстановления всех элементов.

Для проверки функционирования кодера используется тестовый модуль. В нём формируется тестовый набор из четырёх исходных символов, после чего активируется процесс предкодирования. Затем с помощью повторяющихся импульсов иницируется генерация последовательности выходных символов LT-кодера.

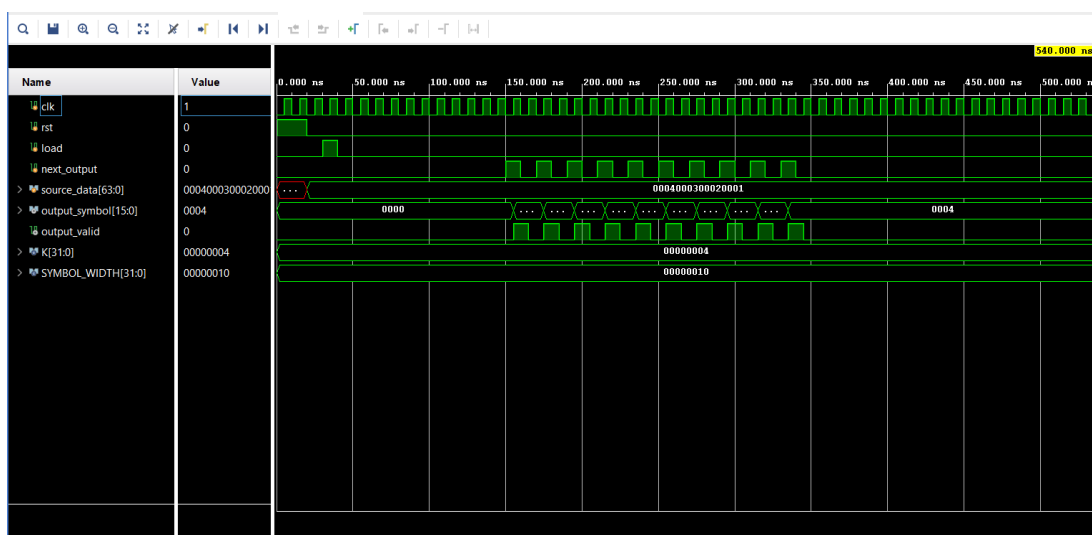


Рис. 1. Временные диаграммы моделирования работы кодера

Каждый сформированный символ выводится в консоль симулятора, что позволяет наблюдать процесс кодирования и убедиться в

корректности работы реализованного алгоритма. На рисунке 1 показаны временные диаграммы сигналов кодера, полученные в результате моделирования.

В результате была получена реализация Raptor-кодера, способная формировать последовательность кодовых символов на основе набора исходных данных. Модульная структура системы обеспечивает возможность её дальнейшей модификации и оптимизации. Разработанная архитектура Raptor-кодера может быть использована при создании средств помехоустойчивой передачи данных, реализованных на базе программируемых логических интегральных схем.

Литература

1. Shokrollahi A. Raptor codes // IEEE Transactions on Information Theory. 2006. №6. С. 2551-2567, doi: 10.1109/TIT.2006.874390.
2. Luby M., Shokrollahi A., Watson M., Stockhammer T. Raptor Forward Error Correction Scheme for Object Delivery. RFC 5053 // Internet Engineering Task Force. 2007. С. DOI: 10.17487/RFC5053.
3. Lázaro C.F., Liva G., Bauch G., Paolini E. Bounds on the Error Probability of Raptor Codes Under Maximum Likelihood Decoding // IEEE Transactions on Information Theory. 2021. №3. С. 1537-1558, doi: 10.1109/TIT.2020.3049061.
4. Richardson T., Urbanke R. Modern Coding Theory. Cambridge: Cambridge University Press, 2008. 816 с.
5. Luby M. LT Codes // Proceedings of the IEEE Symposium on Foundations of Computer Science. 2002. P. 271–280.

A.R. Andreev, A.A. Bondar, O.V. Tikhonova
Russian Technological University – MIREA

DEVELOPMENT OF A RAPTOR ENCODER FOR IMPLEMENTATION ON FPGA

Keywords: *Raptor code, fountain codes, noise-resistant coding, LDPC-precoder, LT-encoder, FPGA.*

Abstract. *In this paper, we consider the implementation of Raptor, an encoder designed for noise-resistant data transmission in communication channels with packet loss. The encoder is implemented in the Verilog language in the Xilinx Vivado environment. The encoder consists of an LDPC precoder and an LT encoder. Raptor coding is described, the architecture of the developed encoder is described, and the features of the algorithm implementation are considered.*

**ВОПРОСЫ ИДЕНТИФИКАЦИИ ПРИ ОРГАНИЗАЦИИ
ХРАНЕНИЯ И ПОИСКЕ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ
С НЕТОЧНО ИЗВЕСТНЫМИ МУЛЬТИМОДАЛЬНЫМИ
АТТРИБУТАМИ В ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЕННЫХ
ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ
СИСТЕМАХ**

Ключевые слова: *территориально-распределенная информационная система, информационно-телекоммуникационная система, информационный объект, идентификация объектов, информационный поиск, критерии идентификации.*

При формировании хранилищ и при решении аналитических задач в территориально распределенных информационно-телекоммуникационных системах (ИТКС), получающих данные из многочисленных источников информации, априори следует считать, что информационные объекты (ИО) имеют искажения. Причинами искажений могут быть ошибки ввода текста оператором, различие вариантов транслитерации при записи фамильно-именной группы (ФИО) и др. Фотографии и другие биометрические данные физических лиц (ФЛ) также следует причислять к искаженным данным, поскольку они могут поступать в систему в разных вариантах оцифровки.

В статье рассмотрены постановки и приведены методики решения задач идентификации искаженных информационных объектов (ИИО) при наличии в ИТКС мультимодальных данных с одинаковыми наборами атрибутов (что обычно бывает, если источники данных однородны) и с разными наборами атрибутов (что характерно для разнородных источников).

Отмечено, что идентификация ИИО с учетом искажений и неполноты значений атрибутов и кластеризация ИИО в хранилище ИТКС принципиально повышают результативность поисково-аналитических запросов, в том числе потому, что поиск событий и связей для кластеров, объединяющих ИИО, соответствующих определенным критериям «похожести», выполняется как для единых ИИО.

Введение. В докладе [1] на конференции СКМП-2021 были рассмотрены особенности обработки потоков информации в многоуровневых территориально-распределенных системах,

получающих из различных источников данные о событиях с участием ИО с неточно известными значениями атрибутов, то есть имеющими различные искажения относительно фактических значений.

ИО являются основными информационными единицами, которыми оперируют ИТКС. Каждый ИО представляет собой совокупность данных, характеризующих некоторую единую сущность. Эти данные являются атрибутами ИО (например, паспортные или биометрические данные физического лица). Данные, характеризующие события, являются атрибутами события (например, время и место события).

Атрибуты ИО формируются по данным различной модальности о событиях, происходящих с ИО в разное время, получаемым от различных внешних систем. Искажения значений атрибутов ИО могут быть вызваны ошибками ввода текста при внесении в систему – источник данных (в том числе фонетические искажения при вводе «со слуха»), различной записью ФИО при выполнении транслитерации (например, при переводе из кириллицы или арабского письма в латиницу), ошибками цифровой обработки заполненных форм документов и др. Фотографии и другие биометрические данные ФЛ также следует причислять к искаженным данным, поскольку они могут поступать в систему в разных вариантах оцифровки. Так, вопросы учета искажений при нейросетевой обработке изображений рассмотрены в работе [2]. К искажениям следует отнести и несоответствие состава и форматов полей данных об одном и том же ИО, поступающих в ИТКС из разных источников. Таким образом, ИО в ИТКС, получающие данные о событиях, априори следует считать информационными объектами с искажениями (ИИО).

Существенное влияние на результативность аналитической работы оказывает организация баз данных (БД) в ИТКС, содержащих разнородную мультимодальную информацию о событиях, поступающую из разных источников [3]. Авторами разработаны [4] методы наполнения БД, реализующие автоматическое связывание «похожих» по заданному критерию ИИО (т. е. кластеризацию – формирование «Сложных ИИО», состоящих из «похожих» ИИО), что очень важно для быстрого выполнения поисково-аналитических запросов и информативности результатов поиска. В этом случае обеспечивается возможность поиска информации о связанных ИИО (составляющих кластер) как об одном объекте с учетом всех значений атрибутов, зафиксированных в системе по данным о событиях, происходивших с ИИО в разное время.

И в задачах поиска объектов, и при пополнении БД ИТКС важно уметь определять, являются ли два и более ИИО с отличающимися атрибутами одним и тем же ИИО по заданным критериям (т. е. выполнять идентификацию ИИО).

В настоящей работе в развитие [1] рассматриваются методы идентификации ИИО с мультимодальными атрибутами.

Под **идентификацией** информационных объектов здесь понимается процесс установления тождественности (идентичности) некоторого информационного объекта другому информационному объекту. Тождественность (идентичность) устанавливается на основании критерия близости атрибутов объектов.

Процесс идентификации ИИО выполняется в ИТКС:

- при поиске ИИО в БД по заданным атрибутам (сравнивается «эталонный» ИИО, заданный значениями атрибутов, с другими ИИО в БД);
- при размещении «нового» ИИО (т.е. ИИО, выделенного из записи о новом событии, поступившей в ИТКС) в БД, а именно: если в БД уже есть такой ИИО, что у него и у «нового» ИИО значения атрибутов в точности совпадают, то «новый» ИИО не вносится в БД во избежание дублирования; иначе выполняется кластеризация: «новый» ИИО добавляется в кластер, состоящий из ИИО, удовлетворяющих заданным критериям «похожести», либо (если нет подходящего кластера) образует новый кластер.

Таким образом, разработка методов идентификации ИИО в ИТКС является актуальной задачей.

Атрибуты каждого ИИО заполняются значениями из данных о событиях.

Пусть X – множество ИИО одного вида (например, физических лиц), D – множество атрибутов ИИО, K – число источников. Каждый ИИО задается набором атрибутов D . Атрибуты $d \in D$ неоднородны: числовые, текстовые, изображения. В общем случае (при получении данных из источников разных типов) наборы атрибутов ИИО различны. При построении алгоритмов идентификации следует учитывать, что некоторые атрибуты могут быть взаимосвязаны.

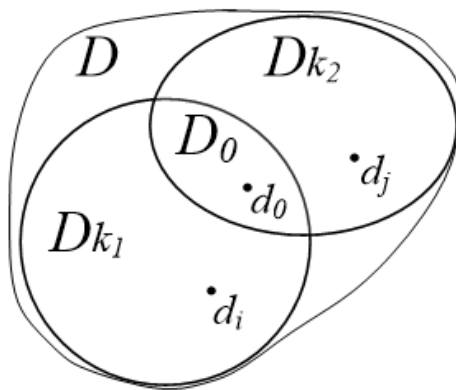


Рис. 1. Пересечение множеств атрибутов информационных объектов

Если число источников K , а набор атрибутов k -го ($1 \leq k \leq K$) источника есть D_k , то $D = \bigcup_{k=1}^K D_k$, а $D_0 = \bigcap_{k=1}^K D_k$, причем D_0 не пусто (как показано на рисунке 1).

Рассмотрим задачи идентификации ИИО в ИТКС в порядке их усложнения:

1) задача идентификации ИИО с **одинаковыми наборами атрибутов** различной модальности (наборы атрибутов могут быть одинаковы, если источники данных однородны).

2) задача идентификации ИИО с **разными наборами атрибутов** различной модальности (наборы атрибутов могут быть разными, если источники данных разнородны).

1. Задача идентификации ИИО с одинаковыми наборами атрибутов.

Постановка задачи

Пусть x_1 и x_2 – ИИО, выделенные из данных о событиях, поступивших из разных источников. Пусть $D(x_1)$ и $D(x_2)$ – наборы атрибутов объектов x_1 и x_2 соответственно и пусть $D(x_1) = D(x_2) = D = \{d_i |_{i=1, \dots, n}\}$ (наборы атрибутов ИИО из разных источников одинаковы, что обычно бывает в случае, если источники однородны). Пусть $\delta_i^1 = d_i(x_1)$ и $\delta_i^2 = d_i(x_2)$ ($i = 1, \dots, n$) – значения атрибутов d_i объектов x_1 и x_2 соответственно.

Нужно определить, являются ли идентичными объекты x_1 и x_2 , где $x_1 \in X$, $x_2 \in X$.

Методика решения задачи

Для принятия решения об идентичности x_1 и x_2 нужно проверить совпадение значений $\delta_i^1 = d_i(x_1)$ и $\delta_i^2 = d_i(x_2)$ ($i = 1, \dots, n$) атрибутов объектов x_1 и x_2 и по результатам проверки сделать один из следующих выводов.

1) Если значения всех атрибутов $d_i \in D$ совпадают: $\forall i : d_i \in D \quad d_i(x_1) = d_i(x_2) = \delta_i$, то объекты x_1 и x_2 являются идентичными: $x_1 = x_2$.

2) В противном случае, – если $\exists i: d_i \in D, d_i(x_1) \neq d_i(x_2)$, – то суждение об идентичности x_1 и x_2 выносится по оценке значения функции $P(x_1, x_2)$, характеризующей близость ИИО x_1 и x_2 ($0 \leq P(x_1, x_2) \leq 1$). По заданному порогу – критическому значению $P_{critical}$ – принимается решение об идентичности x_1 и x_2 : если $P(x_1, x_2) \geq P_{critical}$, то x_1 и x_2 считаются идентичными. В формализованном виде вышеприведенное рассуждение имеет вид:

$$P(x_1, x_2) \geq P_{critical} \Rightarrow x_1 = x_2.$$

Функция $P(x_1, x_2)$ в простейшем случае выражается формулой (средневзвешенное оценок близости по значениям всех атрибутов объекта):

$$P(x_1, x_2) = \sum_{i: d_i \in D} \alpha_i \phi_i(\delta_i^1, \delta_i^2), \quad (1)$$

где α_i – вес атрибута d_i , $\sum_i \alpha_i = 1$;

$\phi_i(\cdot, \cdot)$ – функция оценки близости по атрибуту d_i (область значений каждой функции: $[0,1]$, при этом чем ближе объекты, тем больше значение функции; функции ϕ_i для разных атрибутов могут существенно отличаться друг от друга);

δ_i^1, δ_i^2 – значения атрибута d_i для x_1 и x_2 .

Замечание 1. Для оценки $P(x_1, x_2)$ можно применить более сложное выражение, учитывающее взаимосвязи между атрибутами d_i , оказывающие влияние на веса α_i и функции ϕ_i . В качестве пояснения можно привести пример, когда ИИО представляют собой ФЛ, их атрибуты – паспортные данные. Атрибуту «номер паспорта» целесообразно присвоить достаточно большой вес α , поскольку совпадение номеров паспортов – «сильный аргумент» в пользу того, что ИИО идентичны, даже если в текстовых атрибутах (таких как ФИО) есть различия (возможно, вызванные искажениями). Но, с другой стороны, если значения ФИО совпадают, а номера паспортов разные, то тоже есть большая вероятность идентичности ИИО, так как человек мог просто поменять паспорт. И в этом случае следовало бы приписывать атрибуту «номер паспорта» малый вес. Допустим, что учитывается J ситуаций. Если через α_{ij} обозначить вес i -го атрибута в j -ой ситуации из числа учитываемых, то $P(x_1, x_2)$ оценивается по формуле:

$$P(x_1, x_2) = \max_{1 \leq j \leq J} \left(\sum_{i: d_i \in D} \alpha_{ij} \phi_i(\delta_i^1, \delta_i^2) \right), \text{ причем } \forall j \sum_i \alpha_{ij} = 1. \quad (2)$$

Еще точнее было бы пользоваться не единой функцией близости ϕ_i в различных ситуациях, а ввести набор функций ϕ_{ij} , но это повлекло бы рост объема вычислений, что является препятствием для решения задач поиска ИИО в реальном времени.

Замечание 2. Особо следует рассмотреть вид функций близости ϕ_i для текстовых атрибутов. Для сравнения текстовых атрибутов можно использовать «фонетические ключи» [5], переводящие исходное написание в последовательность знаков. При этом по-разному записывающиеся, но схоже звучащие имена и фамилии имеют одинаковый фонетический ключ. При сравнении значений текстовых атрибутов оценку близости можно вычислять по формуле:

$$P(x_1, x_2) = \max_{1 \leq j \leq J} \left(\sum_{i: d_i \in D} \alpha_{ij} \phi_i(\psi(\delta_i^1), \psi(\delta_i^2)) \right), \quad (3)$$

где ψ есть функция перевода значения атрибута в его фонетический ключ.

Замечание 3. Если в задаче важно избежать ошибок второго рода (принятие различных ИИО за идентичные), то $P_{critical}$ следует выбрать большим, если важнее свести к минимуму ошибки первого рода (принять идентичные ИИО за различные), то величину $P_{critical}$ можно уменьшить.

2. Задача идентификации ИИО с разными наборами атрибутов.

Здесь рассматривается более общий случай, когда в ИТКС поступают мультимодальные данные из разнородных источников, при этом ИИО имеют различные наборы атрибутов. В таких случаях поля данных одного и того же по сути ИИО могут быть различными по составу и содержанию. Например, при покупке железнодорожных (ж/д) билетов граждане РФ чаще всего используют российский паспорт и данные фамильно-именной группы вводят кириллицей (также присутствуют поля: «номер телефона», «адрес электронной почты»), а при покупке авиабилетов поля ФИО заполняются латиницей (независимо от того, используется внутрироссийский или заграничный паспорт), и информация о написании ФИО кириллицей (а также о номере телефона и адресе электронной почты) отсутствует в данных (т. е. в атрибутах ИИО), передаваемых такой системой. При этом достаточно частыми являются искажения атрибутов, вызванные случайными ошибками.

Постановка задачи

Пусть $x_1 \in X$, $x_2 \in X$, но $D(x_1) \neq D(x_2)$, т.е. наборы атрибутов у x_1 и x_2 различны (см. рисунок 2). Нужно определить, являются ли идентичными объекты x_1 и x_2 : $x_1 \in X$, $x_2 \in X$ при условии, что между некоторыми атрибутами ИИО существует функциональная связь: $\exists f : d_m^{(1)}(x_1) \xrightarrow{f} d_m^{(2)}(x_1)$.

Пример постановки задачи. Пусть $x_1 \in X$, $x_2 \in X$, при этом в наборе $D(x_1)$ есть атрибут «имя кириллицей», но нет атрибута «имя латиницей», а в наборе $D(x_2)$ есть атрибут «имя латиницей», но нет атрибута «имя кириллицей». Нужно определить, являются ли идентичными объекты x_1 и x_2 .

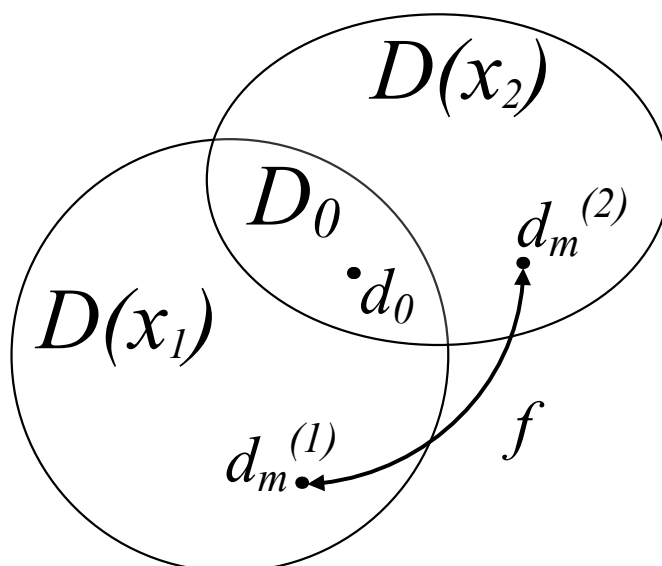


Рис. 2. Пересечение множеств атрибутов информационных объектов при наличии функциональной связи между атрибутами

Методика решения задачи

Если значения атрибутов частично совпадают или близки друг другу, для принятия решения об идентичности x_1 и x_2 нужно вычислить значение функции близости $P(x_1, x_2)$. Если $P(x_1, x_2) \geq P_{critical}$ ($P_{critical}$ – порог), то x_1 и x_2 считаются идентичными. $P(x_1, x_2)$ вычисляется следующим образом.

1. Для атрибутов, входящих в общую область $D_0 = D(x_1) \cap D(x_2)$, оценку $P(x_1, x_2)$ следует вычислять по одной из формул (1) – (3), в зависимости от требований к соотношению точность вычислений / время расчетов для конкретной задачи.

2. Для взаимосвязанных атрибутов в остающихся областях $D(x_1) \setminus D_0$ и $D(x_2) \setminus D_0$ для перевода значения атрибута $d_m^{(1)}(x_1)$ в $d_m^{(2)}(x_1)$ (здесь $d_m^{(1)} \in D(x_1) \setminus D_0$, $d_m^{(2)} \in D(x_2) \setminus D_0$) следует применить функцию f , а для перевода значения атрибута $d_m^{(2)}(x_2)$ в $d_m^{(1)}(x_2)$ применить обратную функцию f^{-1} . После чего вычислить оценку $P(x_1, x_2)$ по одной из формул (1) – (3).

Например, для ИИО, полученных из разных источников, может быть задан только один из двух атрибутов «имя кириллицей», «имя латиницей», но они по определенной процедуре могут друг в друга переводиться.

Рекомендация. Для предотвращения неоднозначности преобразований значений атрибутов при выполнении транслитерации следует использовать фонетические ключи. Тогда получается следующая формула для оценки значения функции близости $P(x_1, x_2)$:

$$P(x_1, x_2) = \max_{1 \leq j \leq J} \left(\sum_{i: d_i \in D_0} \alpha_{ij} \phi_i(\psi(\delta_i^1), \psi(\delta_i^2)) + \sum_{m: d_m \in D_M} \alpha_{mj} \phi_m(\psi(\delta_m^1), \psi(\delta_m^2)) \right), \quad (4)$$

где

$$D_M = \{d_m^{(2)} : \exists f, f(d_m^{(1)}) \Rightarrow d_m^{(2)}, \text{ где } d_m^{(1)} \in D_1 \setminus D_0, d_m^{(2)} \in D_2 \setminus D_0\}.$$

При этом для практической настройки алгоритма расчета при принятых процедурах транслитерации f и взятия фонетического ключа ψ необходимо определить с помощью экспертов значения весов α и оценок близости ϕ , а также критического значения $P_{critical}$.

Заключение

В работе введена концепция, состоящая в том, что в реальных условиях все информационные объекты, данные о которых поступают и сохраняются в ИТКС, целесообразно считать информационными объектами с искажениями (введен термин: «искаженный информационный объект» – ИИО).

Для рассматриваемого класса территориально распределенных ИТКС, например для региональных и межрегиональных информационных центров Фонда пенсионного и социального страхования Российской Федерации, получающих мультимодальные данные из разнородных источников информации и оперирующих информационными объектами со случайными или намеренно внесенными искажениями, приведены постановки и методики решения

задач автоматической идентификации ИИО в БД для двух вариантов (одинаковый и различный набор атрибутов ИИО).

Разработанные технологии организации информационных хранилищ с автоматическим связыванием в БД ИТКС «похожих» по определенным критериям ИИО (кластеризацией) и методы идентификации ИИО с учетом искажений значительно повышают скорость выполнения и результативность поисково-аналитических запросов операторов. Так, поиск связей между «Сложными ИИО» осуществляется как между едиными ИИО с учетом значений атрибутов всех ИИО, входящих в каждый «Сложный ИИО».

Литература

1. Бирюкова Т.К., Гершкович М.М. Особенности компьютерной обработки данных с неточно известными параметрами в многоуровневых территориально-распределенных информационно-телекоммуникационных системах // Системы компьютерной математики и их приложения (СКМП-2021): материалы XXII Международной научной конференции (Смоленск, 28–29 мая 2021). Смоленск: Изд-во СмолГУ, 2021. Вып. 22. С. 27–36.
2. Задача классификации в условиях искаженных причинно-следственных связей / А.А. Грушо [и др.] // Системы и средства информатики. 2023. Т. 33, № 1. С. 59-67.
3. Зацаринный А.А., Ионенков Ю.С. Оценка эффективности информационно-телекоммуникационных систем. М.: НИПКЦ Восход-А, 2020. 120 с.
4. Гершкович М.М., Бирюкова Т.К., Сеницин В.И. О возможностях распознавания информационных объектов с неточно известными атрибутами и биометрическими характеристиками в информационных системах, получающих данные из разнородных источников // Сборник материалов XVI Международной научно-технической конференции «Опτικο-электронные приборы и устройства в системах распознавания образов и обработки изображений. Распознавание – 2021» (Курск, 14-17 сентября 2021 г.); Юго-Зап. гос. ун-т. Курск, 2021. С. 90-92.
5. Выхованец В.С., Ду Ц., Сакулин С.А. Обзор алгоритмов фонетического кодирования // УБС. 2018. Вып. 73. С. 67–94.

T.K. Biryukova, M.M. Gershkovich, V.I. Sinitsin
*Federal Research Center «Computer Science and Control» of the
Russian Academy of Sciences*

IDENTIFICATION ISSUES IN ORGANIZING STORAGE AND SEARCHING INFORMATION OBJECTS WITH NOT EXACTLY KNOWN MULTIMODAL ATTRIBUTES IN GEOGRAPHICALLY DISTRIBUTED INFORMATION-TELECOMMUNICATION SYSTEMS

Keywords: *area spread information system, information-telecommunication system, informational object, identification of objects, informational search, identification criteria*

Abstract. *When creating data stores and solving analytical problems in geographically distributed information-telecommunication systems (ITCS) that receive data from multiple sources, it should be assumed a priori that information objects (IO) are distorted. Distortions can be caused by operator input errors, differences in transliteration when recording family names, and other factors. Photographs and other biometric data of individuals should also be considered distorted data, as they may be entered into the system in different digitization formats.*

This article examines the formulations and methods for solving problems of identifying distorted information objects (DIO) in the presence of multimodal data in the ITCS with identical sets of attributes (which typically occurs if the data sources are homogeneous) and with different sets of attributes (which is typical for heterogeneous sources).

It is noted that the identification of DIOs considering distortions and incompleteness of attribute values and the clustering of DIOs in the ITCS store radically improve the effectiveness of search and analytical queries, not least because the search for events and connections for clusters of meeting the criteria of similarity DIOs is performed as for single DIOs.

М.А. Ганин

Российский технологический университет МИРЭА

УДК 621.391

ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ЗАКА И ОПЕРАТОРНАЯ МОДЕЛЬ ЛИНЕЙНОГО НЕСТАЦИОНАРНОГО КАНАЛА: ИНТЕРПРЕТАЦИЯ В ДОМЕНЕ «ЗАДЕРЖКА-ДОПЛЕР»

Ключевые слова: *дискретное преобразование Зака, домен задержка–доплер, скрученная свертка, унитарный изоморфизм, разреженность, трёхмерная визуализация, интегрированная сенсорика и связь.*

В данной статье дискретное преобразование Зака рассматривается как унитарный изоморфизм между пространством временных сигналов и решёткой «задержка-доплер». В рамках данного изоморфизма оператор линейного нестационарного канала принимает форму скрученной свертки, сохраняющей разреженность физического канала. Приводятся три сценария трёхмерной визуализации представления в домене «задержка-доплер», устанавливающие связь геометрических паттернов с вычислительной сложностью выравнивания канала. Предложенная модель ориентирована на реализацию в среде программирования Python и применима в системах интегрированной сенсорики и связи.

Введение. Современные системы компьютерной математики (например, язык программирования Python с библиотеками NumPy и SciPy) оперируют унитарными преобразованиями как средством смены координат, гарантируя сохранение энергии и обратимость алгоритмов [2, с. 96; 3, с. 33]. Если для линейных стационарных систем таким стандартом является дискретное преобразование Фурье, то для линейных нестационарных каналов, характеризующихся двойным рассеянием по времени и частоте, Фурье-координаты не обеспечивают разреженного представления [4, с. 24, 106]. Домен «задержка-доплер» параметризует канал непосредственно через физические величины — задержку и доплеровский сдвиг, что делает его естественным языком описания мобильной связи и радиолокации [1, с. 105; 2, с. 53].

В контексте развития систем интегрированной сенсорики и связи и стандартов поколения 6G возрастает потребность в методах, способных эффективно обрабатывать сигналы в условиях высоких доплеровских сдвигов [4, с. 15; 2]. Дискретное преобразование Зака является недостающим звеном между физической моделью линейного нестационарного канала и вычислительным аппаратом систем компьютерной математики, обеспечивая прямое отображение временного сигнала в домен «задержка-доплер» в единой операторной форме без промежуточного перехода в частотно-временную область [2, с. 100; 3, с. 24–25]. В данной работе предлагается детальное математическое описание оператора канала в терминах алгебры Гейзенберга на конечной решётке [5, с. 851; 7, с. 1].

1. Математическое описание дискретного преобразования Зака.

Рассмотрим дискретные сигналы конечной длины. Введём решётку «задержка-доплер», где параметры дискретизации определяются физическими характеристиками канала [2, с. 53; 4, с. 52]. Гильбертово пространство сигналов обозначим как $\mathcal{H}$. Данный подход к моделированию сигналов непосредственно в домене «задержка-доплер» был впервые предложен в работе [7, с. 1].

Математически преобразование Зака реализует **унитарный изоморфизм** гильбертова пространства временных сигналов $\mathcal{H}$ на пространство функций на решётке «задержка-доплер». Это означает взаимно однозначное соответствие между отсчётами сигнала и коэффициентами в домене «задержка-доплер» с сохранением скалярного произведения [3, с. 33; 5, с. 853].

Для одномерного временного сигнала $x[n]$, где $n = 0, 1, \dots, N - 1$, прямое дискретное преобразование Зака определяется следующим выражением:

$$X_Z[k, l] = \sum_{m=0}^{M-1} x[n + mN] e^{-j2\pi \frac{l}{L} m} \quad (1)$$

где $X_Z[k, l]$ – коэффициенты преобразования Зака в домене «задержка-доплер»; n – индекс временного отсчета внутри блока ($0 \leq n < N$); k – индекс задержки; l – индекс доплеровского сдвига; N – число доплеровских (временных) интервалов; L – число задержек (частотных поднесущих); M — коэффициент периодичности решётки; j – мнимая единица [5, с. 859, ф-ла 30].

Обратное преобразование, позволяющее восстановить временной сигнал по его представлению в домене «задержка-доплер», имеет вид:

$$x[n] = \frac{1}{L} \sum_{l=0}^{L-1} X_Z[n \bmod N, l] e^{j2\pi \frac{l}{L} n} \quad (2)$$

Преобразование является унитарным, что гарантирует сохранение энергии сигнала при переходе между доменами [4, с. 323; 5, с. 853]:

$$\sum_n |x[n]|^2 = \frac{1}{L} \sum_{k,l} |X_Z[k, l]|^2 \quad (3)$$

Ключевое свойство квазипериодичности (свойство Зака) записывается следующим образом:

$$X_Z[k + N, l] = X_Z[k, l] e^{j2\pi \frac{l}{L}} \quad (4)$$

Это свойство отражает циклическую структуру дискретной решётки и критически важно для построения операторных моделей [3, с. 31, ф-ла 2.22; 4, с. 317, ф-ла 8.B.2].

Важно отметить, что предсказуемость входу-выходного соотношения в домене «задержка-доплер» достигается в **кристаллическом режиме**, когда произведение параметров решётки удовлетворяет условию $\tau_p \nu_p = 1$ [4, с. 58, 135]. Именно в этом режиме наблюдаются чёткие «иглы» на трёхмерных графиках (сценарий А), что соответствует максимальной разреженности оператора канала [4, с. 20, 61].

2. Операторная модель линейного нестационарного канала.

Применим описанный выше аппарат для моделирования канала. Физические свойства канала описываются **функцией рассеяния (спред-функцией) $S(\tau, \nu)$** , которая характеризует комплексный коэффициент отражения пути с задержкой τ и доплеровским сдвигом ν [6, с. 74–75]. В

дискретной решётке она принимает вид матрицы H_{DD} [4, с. 195, рис. 5.6]. Канал полностью описывается следующим выражением:

$$H = \iint S(\tau, \nu) T_{\tau, \nu} d\tau d\nu, \quad (5)$$

где H – оператор канала; $S(\tau, \nu)$ – функция рассеяния, зависящая от задержки τ и доплеровского сдвига ν ; $T_{\tau, \nu}$ – оператор сдвига по времени и частоте [4, с. 79; 7, с. 1].

Входной сигнал x и выходной сигнал y связаны оператором скрученной свертки ($\odot$):

$$Y_z[k, l] = (S \odot X_z)[k, l] = \sum_{k', l'} S[k', l'] X_z[k - k', l - l'] e^{j2\pi \frac{(k-k')l}{L}} \quad (6)$$

В формуле (6) индекс k соответствует оси задержки (горизонтальная ось на рис. 2–4), индекс l – оси доплера (горизонтальная ось, перпендикулярная задержке). Разности индексов $(k - k')$ и $(l - l')$ понимаются по модулю N и L соответственно, что следует из свойства квазипериодичности (4) и отражает циклическую структуру дискретной решётки (тор) [4, с. 325; 5, с. 851]. Фазовый множитель учитывает некоммутативность сдвигов по времени и частоте, характерную для алгебры Гейзенберга [4, с. 319, ф-ла 8.В.9]. В отличие от квадратичных распределений (таких как распределение Вигнера – Вилля), линейность и унитарность дискретного преобразования Зака исключают появление перекрёстных членов и гарантируют обратимость преобразования, что является критически важным свойством для численных алгоритмов выравнивания канала [3, с. 26; 4, с. 24].

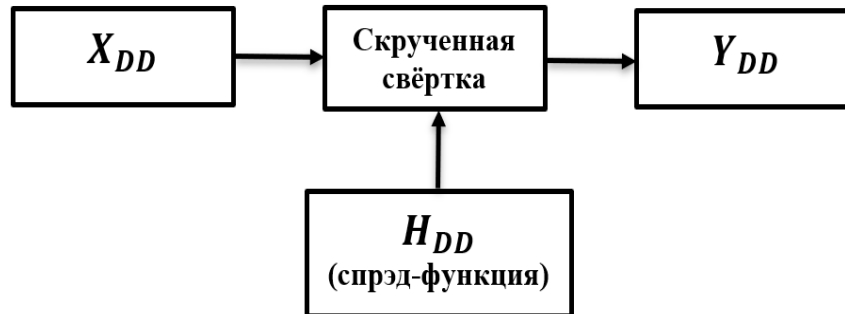


Рис. 1. Блок-схема оператора в домене «задержка-доплер»

Ассоциативность операции скрученной свертки позволяет каскадировать фильтры и каналы в домене «задержка-доплер» без потери точности, что является математическим преимуществом дискретного преобразования Зака перед квадратичными частотно-

временными распределениями [4, с. 81, кор. 2.1; 4, с. 319]. Структура оператора иллюстрирована на рисунке 1.

3. Трёхмерная визуализация и разреженность.

Геометрия модуля передаточной функции $|H|$ (ось Z – амплитуда) непосредственно характеризует разреженность оператора и сложность выравнивания канала [4, с. 68, ф-ла 2.94; 4, с. 102]. Для наглядной демонстрации свойств дискретного преобразования Зака были построены трёхмерные модели распределения энергии канала в решётке «задержка-доплер», как показано на рисунках 2–4.

Сценарий А: Бинированные пути («Иглы»).

Условия: целочисленные задержки и доплер.

Геометрия: изолированные дельта-пики на трёхмерном графике (ось X – задержка, ось Y – доплер, ось Z – $|H|$) [4, с. 41, рис. 2.14; 6, с. 108, рис. 9.3].

Вычислительный аспект: максимальная разреженность матрицы канала – число ненулевых элементов равно числу путей [4, с. 20, 102; 7, с. 1].

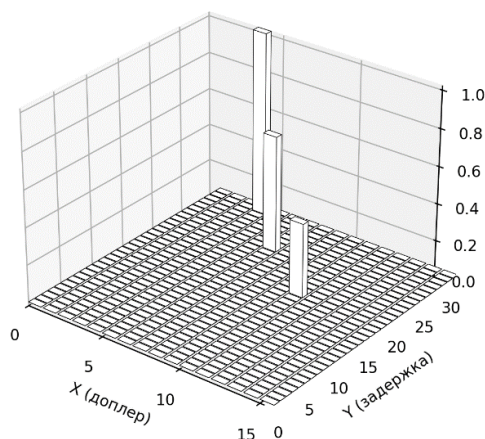


Рис. 2. Трёхмерный график для сценария А (ось Z – амплитуда)

Сценарий В: Дробный доплер («Холмы/гребни»).

Условия: целочисленная задержка, дробный доплер.

Геометрия: «гребень» вдоль оси доплера, обусловленный спектральной утечкой из-за несовпадения частоты пути с шагом [2, с. 77, рис. 4.16; 4, с. 41, рис. 2.14-b].

Вычислительный аспект: разреженность резко снижается — энергия одного пути распределяется по множеству соседних доплеровских ячеек (функция Дирихле), что требует применения оконных функций или итеративных алгоритмов детектирования [2, с. 86; 4, с. 102].

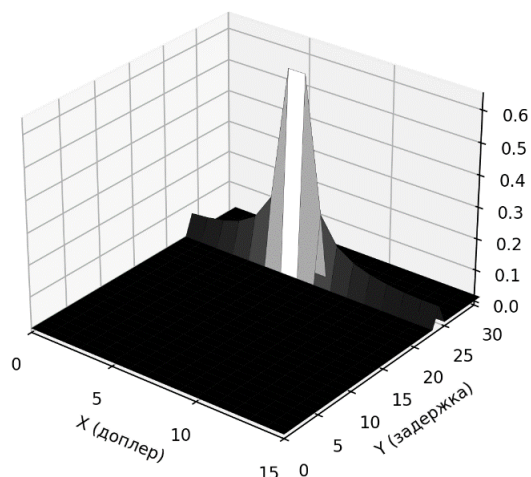


Рис. 3. Трёхмерная поверхность для сценария В (ось Z – амплитуда)

Сценарий С: Сдвиг несущей частоты.

Условия: целочисленная задержка, остаточный сдвиг несущей частоты.

Геометрия: систематический сдвиг всего частотно-временного паттерна в домене «задержка-доплер» вдоль доплеровской оси [2, с. 86; 6, с. 126, рис. 10.5].

Вычислительный аспект: возможность детектирования и оценки сдвига несущей частоты непосредственно из визуального паттерна (структуры функции рассогласования) с последующей компенсацией на этапе предварительной обработки [2, с. 86; 4, с. 292–294].

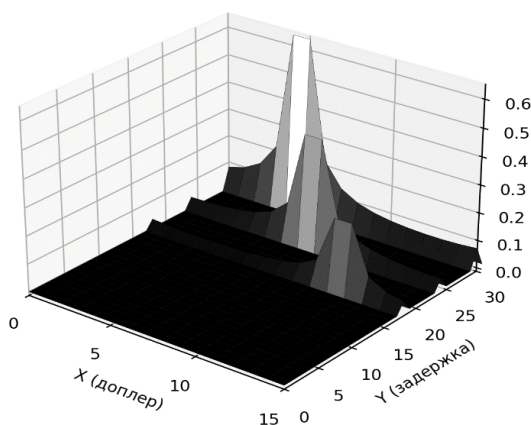


Рис. 4. Трёхмерная поверхность для сценария С (ось Z – амплитуда)

Заключение

1. Дискретное преобразование Зака является унитарным изоморфизмом, переводящим временной сигнал в координаты «задержка-доплер» [3, с. 33; 5, с. 853]. При численной реализации унитарность сохраняется с точностью до согласованного выбора

параметров дискретизации решётки и защитных интервалов [2, с. 53; 4, с. 311].

2. Оператор линейного нестационарного канала в домене задержка–доплер имеет форму скрученной свертки – замкнутой алгебраической структуры (алгебра Гейзенберга на конечной решётке), разреженность которой определяется физическим числом путей распространения сигнала [4, с. 24, 78; 7, с. 1].

3. Трёхмерная визуализация решётки в домене «задержка-доплер» (ось Z – амплитуда) позволяет интерпретировать неидеальности канала как детерминированные геометрические паттерны: «иглы» (максимальная разреженность в кристаллическом режиме), «холмы» (спектральная утечка при дробном доплере) и «сдвиги» (систематический сдвиг из-за расстройки несущей частоты) [2, с. 77, рис. 4.16; 4, с. 147].

4. Предложенная модель непосредственно реализуема в системах компьютерной математики (Python, библиотеки NumPy и SciPy) через встроенные функции быстрого преобразования Фурье [4, с. 311, 316]. Структура скрученной свертки допускает применение разреженных решателей и итеративных методов, что востребовано в задачах интегрированной сенсорики и связи, где единый сигнал используется одновременно для высокоскоростной передачи данных и прецизионного радиолокационного зондирования [4, с. 18, 335; 4, гл. 10].

Литература

1. Ширман Я.Д., Манжос В.Н. Теория и техника обработки радиолокационной информации на фоне помех. М.: Радио и связь, 1981. 416 с.
2. Hong Y., Thaj T., Viterbo E. Delay-Doppler Communications: Principles and Applications // Academic Press, 2022. Ch. 434 p.
3. Janssen A. The Zak transform: A signal transform for sampled time-continuous signals // Philips Journal of Research. 1988. Vol. 43, No. 1. P. 23–69.
4. Mohammed S.K., Hadani R., Chockalingam A. OTFS Modulation: Theory and Applications. Wiley, 2024. 448 p.
5. Bölcskei H., Hlawatsch F. Discrete Zak transforms, polyphase transforms, and applications // IEEE Transactions on Signal Processing. 1997. Vol. 45, No. 4. P. 851–866.
6. Варакин Л.Е. Теория сложных сигналов. М.: Советское радио, 1970. 376 с.
7. Hadani R. et al. Orthogonal Time Frequency Space Modulation. Proc. IEEE WCNC. 2017. P. 1–6.

M.A. Ganin

Russian Technological University – MIREA

DISCRETE ZAK TRANSFORM AND OPERATOR MODEL OF LINEAR TIME-VARYING CHANNEL: INTERPRETATION IN DELAY–DOPPLER DOMAIN

Keywords: *discrete Zak transform, delay-Doppler domain, twisted convolution, unitary isomorphism, sparsity, 3D visualization, integrated sensing and communication.*

Abstract. *The paper considers the Discrete Zak Transform (DZT) as a unitary isomorphism between the space of time signals and the «delay-Doppler» (DD) lattice. Within this isomorphism, the linear time-varying (LTV) channel operator takes the form of a twisted convolution, preserving the sparsity of the physical channel. Three scenarios of 3D visualization of the DD representation are presented, establishing a connection between geometric patterns and the computational complexity of channel equalization. The proposed model is oriented for implementation in Python programming environment.*

В.Р. Кристалинский

*Смоленский государственный университет,
ВА ВПВО ВС РФ*

УДК 519.688

О РЕАЛИЗАЦИИ АЛГОРИТМА ПРИБЛИЖЕННОГО ВЫЧИСЛЕНИЯ ИНТЕГРАЛА ТИПА КОШИ В СИСТЕМЕ WOLFRAM MATHEMATICA

Ключевые слова: *интеграл типа Коши, система Wolfram Mathematica.*

В статье предлагается программная реализация алгоритма приближенного вычисления интеграла типа Коши в системе Wolfram Mathematica, приводится пример. Рассмотрена интеграция этой реализации с приложением на языке C#.

Пусть L – гладкий замкнутый или незамкнутый контур, целиком расположенный в конечной части плоскости; τ – комплексная координата его точек и $f(\tau)$ – непрерывная функция точек контура. Тогда интеграл вида (см., например, [1, с. 19])

$$\Phi(\tau) = \frac{1}{2\pi i} \int_L \frac{f(\tau)}{\tau - z} d\tau$$

называется интегралом типа Коши (см., например, [1, с. 19]), функция $f(\tau)$ называется его плотностью, а $\frac{1}{\tau - z}$ – ядром.

В диссертации [2] и в статьях [3]–[4] нами был предложен приближенный метод вычисления интегралов типа Коши, пригодный

для широкого класса контуров L и плотностей $f(\tau)$. В указанном методе интеграл типа Коши аппроксимируется рациональной функцией в замкнутой области.

Суть предложенного нами алгоритма в случае замкнутого контура заключается в следующем.

Пусть односвязная область D^+ ограничена замкнутым гладким контуром L и содержит внутри себя начало координат. Будем предполагать, что контур L допускает параметризацию, задаваемую функциями, имеющими производные не менее третьего порядка по дуге. Плотность $f(\tau)$ предполагается не менее чем дважды дифференцируемой на контуре L .

Рассмотрим функцию $w = \varphi(z)$, осуществляющую конформное отображение области D^+ на единичный круг, причем $\varphi(0) = 0$, а $z = \psi(w)$ – обратная к ней функция. В силу известных результатов В.И. Смирнова [5] функция $\psi(w)$ не менее чем дважды дифференцируема на единичной окружности. Тогда функция $F(w) = f(\psi(w))$ будет принадлежать классу $C_m(|w|=1)$, $m \geq 2$, где $C_m(|w|=1)$ – класс функций, m раз дифференцируемых на единичной окружности.

Пусть $S_n(w)$ – n -я частичная сумма ряда Фурье функции $F(w)$:

$$S_n(w) = \sum_{k=-n}^n c_k w^k.$$

Рассмотрим интеграл типа Коши

$$\frac{1}{2\pi i} \int_L \frac{f(\tau) d\tau}{\tau - z}. \quad (1)$$

Пусть $\Phi^+(z)$ – функция, определенная этим интегралом в области D^+ . Заменяем плотность $f(\tau)$ на $S_n(\varphi(\tau))$ и вычислим интеграл.

$$\begin{aligned} & \frac{1}{2\pi i} \int_L \frac{S_n(\varphi(\tau))}{\tau - z} d\tau. \\ S_n(\varphi(t)) &= \sum_{k=-n}^n c_k [\varphi(t)]^k = \frac{\sum_{k=-n}^n c_k [\varphi(t)]^{n+k}}{[\varphi(t)]^n}. \end{aligned}$$

Эта функция является граничным значением аналитической функции, имеющей в точке 0 единственный полюс порядка n . Следовательно,

$$\Phi_n^+(z) = \frac{\sum_{k=-n}^n c_k [\varphi(z)]^{n+k}}{[\varphi(z)]^n} - L_n(z),$$

где $L_n(z)$ – главная часть ряда Лорана уменьшаемого.

В диссертации [2] и в статье [4] описана реализация указанного выше алгоритма в системе Mathcad, использующая функцию $L(z)$, полученную с помощью системы Maple. Однако эта компьютерная реализация обладает рядом недостатков. В частности, она не позволяет находить максимальное значение абсолютной величины разности между точным и приближенным значением интеграла типа Коши на контуре в случае, когда точное значение интеграла типа Коши известно, например, если плотность интеграла есть граничное значение аналитической функции. Этого недостатка лишена предлагаемая в настоящей статье реализация данного алгоритма в системе Mathematica. Кроме того, на ее основе можно построить программу на языке C# в стандартном интерфейсе приложения Windows, включающую в себя эту реализацию.

Далее будет рассмотрен тот же пример, что и в диссертации [2]. Приведем соответствующую программу в системе Wolfram Mathematica.

Требуется вычислить интеграл типа Коши с плотностью $\sin z$, где контуром интегрирования является эллипс $x^2 + y^2 - 1 - \frac{3}{4}(x^2 - y^2) = 0$.

Для рассматриваемого эллипса

$$l=0.2$$

Функция, дающая конформное отображение круга на эллипс

$$\begin{aligned} z1[w_] &= (1 - 1/8 * l^2 + 3/128 * l^4) * w \\ z2[w_] &= 1 + (1/2 * l - 1/4 * l^3 + 3/32 * l^5) * w^2 + (1/2 * l^2 - \\ & 9/16 * l^4) * w^4 + (5/8 * l^3 - 9/8 * l^5) * w^6 + 7/8 * l^4 * w^8 + 21/16 * l^5 * w^{10} \\ z[w_] &= z1[w] * z2[w] \end{aligned}$$

$$0.995038 \quad w \quad (1 + 0.09803 \quad w^2 + 0.0191 \quad w^4 + 0.00464 \quad w^6 + 0.0014 \quad w^8 + 0.00042 \quad w^{10})$$

Полуоси эллипса

$$a1 = 1/\text{Sqrt}[1-l]$$

$$b1 = 1/\text{Sqrt}[1+l]$$

Функция, дающая конформное отображение эллипса на круг.

$$w[zz]=zz-1/2*zz^3*1+(1/8*zz+1/4*zz^5)*1^2+(1/16*zz^3-1/8*zz^7)*1^3+(-1/32*zz^5+1/16*zz^9-1/128*zz)*1^4+(-1/64*zz^3+1/64*zz^7-1/32*zz^{11}+1/256*zz^2-1/256*zz^4)*1^5+(-25/256*zz-43/128*zz^5-29/64*zz^9-1/128*zz^2+1/256*zz^6)*1^6*1/2$$

Строим функцию $F(w)$

$$F[w_]=\text{Sin}[z[w]]$$

Вычисляем коэффициенты ряда Фурье этой функции

$$\text{For}[n=0,n\leq 10,n++,b[n]=\text{NIntegrate}[(F[\text{Cos}[f]+I*\text{Sin}[f]])*(\text{Cos}[n*f]+I*\text{Sin}[n*f])/(2*\text{Pi}),\{f,0,2*\text{Pi}\}]]$$

$$\text{For}[n=0,n\leq 10,n++,a[n]=\text{NIntegrate}[(F[\text{Cos}[f]+I*\text{Sin}[f]])*(\text{Cos}[n*f]-I*\text{Sin}[n*f])/(2*\text{Pi}),\{f,0,2*\text{Pi}\}]]$$

Строим функцию

$$f2[zz_]=\text{Sum}[a[n]*(w[zz])^n+b[n]/((w[zz]))^n,\{n,1,10\}];$$

Строим главную часть ряда Лорана этой функции

$$\text{Series}[f2[zz],\{zz,0,-1\}]$$

$$L[zz_]=\text{Normal}[\text{Series}[f2[zz],\{zz,0,-1\}]]$$

Окончательно получаем

$$FF[zz_]=f2[zz]-L[zz]$$

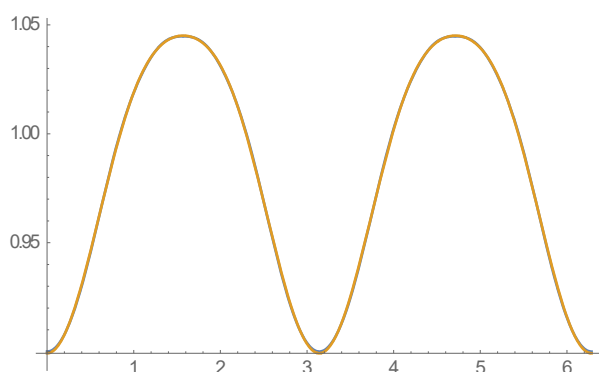


Рис. 1

Строим графики модулей полученного приближения для функции $\Phi^+(z)$ и ее точного значения – функции $\sin(z)$ на эллипсе (см. рис. 1):

```
Plot[{Abs[FF[a1*Cos[t1]+I*b1*Sin[t1]]],Abs[Sin[a1*Cos[t1]+I*b1*Sin[t1]]]}, {t1,0,2*Pi}]
```

Как видно из рисунка 1, графики практически совпадают.

Вычисляем максимальную разность между приближенным и точным значениями функции $\Phi^+(z)$ на эллипсе

```
NMaximize[Abs[FF[a1*Cos[t1]+I*b1*Sin[t1]]]-Abs[Sin[a1*Cos[t1]+I*b1*Sin[t1]]],t1]
{0.000747721,{t1→7.10552*10^-8}}
```

Как видим, погрешность полученного приближения составляет 0,07%.

На основе описанного примера аналогично тому, как это было описано в статьях [6] – [8], разработано интегрированное приложение на языке C#, реализующее данный алгоритм. Код программы для простейшего случая, когда все исходные данные вводятся в тексте программы, приведен ниже:

```
mathKernel1.ResultFormat =
Wolfram.NETLink.MathKernel.ResultFormatType.InputForm;
mathKernel1.Compute("l = 0.2");
mathKernel1.Compute("z1[w_] = (1 - 1/8*l^2 + 3/128*l^4)*w");
mathKernel1.Compute("z2[w_] = 1 + (1/2*l - 1/4*l^3 +
3/32*l^5)*w^2 + (1/2*l^2 - 9/16*l^4)* w ^ 4 + (5 / 8 * l ^ 3 - 9 / 8 * l ^ 5) *
w ^ 6 + 7 / 8 * l ^ 4 * w ^ 8 + 21 / 16 * l ^ 5 * w ^ 10");
mathKernel1.Compute("z[w_] = z1[w]*z2[w]");
mathKernel1.Compute("a1 = 1/Sqrt[1 - l]");
mathKernel1.Compute("b1 = 1/Sqrt[1 + l]");
mathKernel1.Compute("w[zz] = zz - 1 / 2 * zz ^ 3 * l + (1 / 8 * zz + 1
/ 4 * zz ^ 5) * l ^ 2 + (1 / 16 * zz ^ 3 - 1 / 8 * zz ^ 7) * l ^ 3 + (-1 / 32 * zz ^ 5
+ 1 / 16 * zz ^ 9 - 1 / 128 * zz) * l ^ 4 + (-1 / 64 * zz ^ 3 + 1 / 64 * zz ^ 7 - 1 /
32 * zz ^ 11 + 1 / 256 * zz ^ 2 - 1 / 256 * zz ^ 4) * l ^ 5 + (-25 / 256 * zz -
43 / 128 * zz ^ 5 - 29 / 64 * zz ^ 9 - 1 / 128 * zz ^ 2 + 1 / 256 * zz ^ 6) * l ^
6 * 1 / 2");
mathKernel1.Compute("F[w_] = Sin[z[w]]");
mathKernel1.Compute("b = Table[NIntegrate[(F[Cos[f] + I * Sin[f]])
* (Cos[(n - 1) * f] + I * Sin[(n - 1) * f]) / (2 * Pi), { f, 0, 2 * Pi}], { n, 1,
11}]);
```

```

mathKernel1.Compute("a = Table[NIntegrate[(F[Cos[f] + I * Sin[f]])
* (Cos[(n - 1) * f] - I * Sin[(n - 1) * f]) / (2 * Pi), {f, 0, 2 * Pi}], {n, 1,
11}]]");
mathKernel1.Compute("f2[zz_] = Sum[a[[n]]*(w[zz])^(n - 1), {n, 1,
11}] + Sum[b[[n]] / ((w[zz]) ^ (n - 1)), {n, 2, 11}]]");
mathKernel1.Compute("Series[f2[zz], {zz, 0, -1}]]");
mathKernel1.Compute("L[zz_] = Normal[Series[f2[zz], {zz, 0, -
1}]]");
mathKernel1.Compute("FF[zz_] = f2[zz] - L[zz]");
mathKernel1.Compute("NMaximize[Abs[FF[a1 * Cos[t1] + I * b1 *
Sin[t1]]] - Abs[Sin[a1 * Cos[t1] + I * b1 * Sin[t1]]], t1]");
lblRes.Text = mathKernel1.Result.ToString();

```

Результат работы программы представлен на рисунке 2. На форму выводится абсолютная погрешность полученного приближения для интеграла типа Коши.

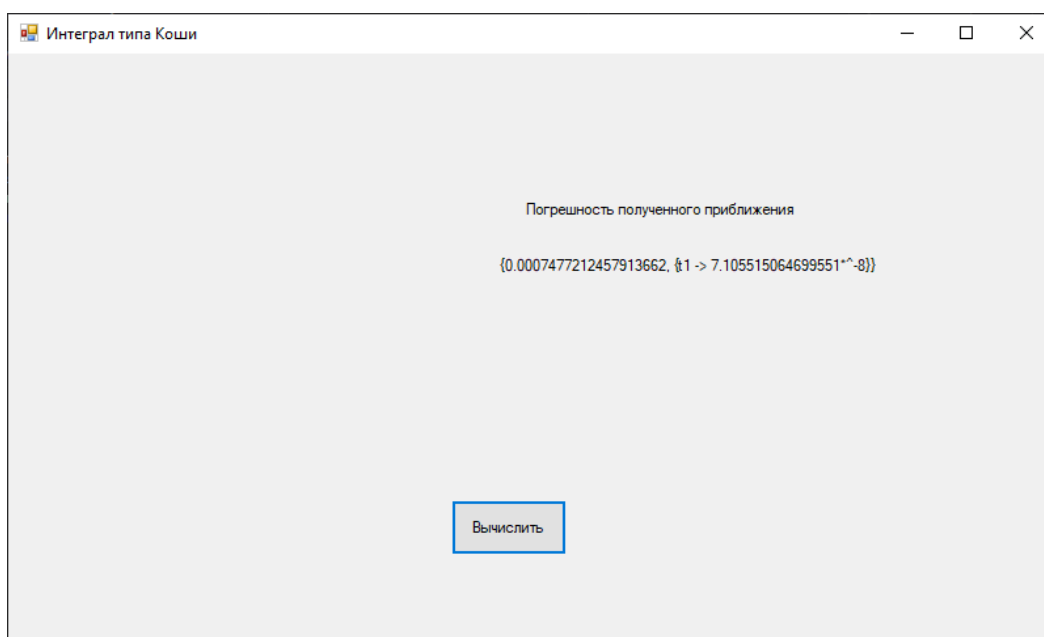


Рис. 2. Форма с результатами вычислений

Приведенная выше программа носит несколько упрощенный характер и лишь иллюстрирует возможность создания интегрированного приложения на основе предложенного метода приближенного вычисления интеграла типа Коши. Ее можно дополнить возможностью ввода пользователем на форму, например, числа членов ряда Фурье, используемого в приложении и выводом графиков, аналогично тому, как это описано нами в статье [9].

Литература

1. Гахов Ф.Д. Краевые задачи. М.: Наука, 1977. 640 с.
2. Кристалинский В.Р. Приближенные методы решения краевых задач типа Гильберта и типа Римана для бианалитических функций: специальность 01.01.01 "Вещественный, комплексный и функциональный анализ": дис. ... канд. физ.-мат. наук. Смоленск, 2001. 114 с.
3. Kristalinskii V.R. An approximate method for computation of Cauchy-type integrals // Proceedings The 20-th Summer School "Applications of Mathematics to Engineering". Varna, 1994.
4. Кристалинский В.Р. Об одном методе приближенного вычисления интегралов типа Коши // Полианалитические функции: граничные свойства и краевые задачи. Смоленск, 1997. С. 34-38.
5. Смирнов В.И. Курс высшей математики: Т.3, часть 2. М.: Наука, 1974. 672 с.
6. Кристалинский В.Р., Лютикас П.Л. О построении интерфейса между документом Wolfram Mathematica и проектом Visual Studio на языке C# // Системы компьютерной математики и их приложения. 2024. Вып. 25. С. 26-31.
7. Черный С.Н., Кристалинский В.Р. Построение комплекса программ, моделирующих наведение зенитных управляемых ракет // Современные информационные технологии и ИТ-образование. 2024. Т. 20, № 2. С. 477-487.
8. Кристалинский В.Р., Рыкалова М.Е. Вычисление вероятности попадания ракеты в круг заданного радиуса в интегрированном приложении Wolfram Mathematica и C# // Современная педагогика и научные исследования в образовательной организации высшего образования: сборник докладов очно-заочной научно-методической конференции, Кострома, 16 февраля 2025 года. Кострома: Военная академия радиационной, химической и биологической защиты им. Маршала Советского Союза С.К. Тимошенко, 2025. С. 158-166.
9. Кристалинский В.Р. О программе для приближенного решения краевой задачи Гильберта для аналитических функций // Таврический вестник информатики и математики. 2025. №3(68). С. 86–95.

V.R. Kristalinskiy

Smolensk State University

*Russian Federation Armed Forces Army Air Defense Military Academy
named after Marshal of the Soviet Union A.M. Vasilevsky*

ON THE IMPLEMENTATION OF AN ALGORITHM FOR APPROXIMATE CALCULATION OF THE CAUCHY-TYPE INTEGRAL IN THE WOLFRAM MATHEMATICA SYSTEM

Keywords: *the Cauchy-type integral, the Wolfram Mathematica system.*

Abstract. *This paper considers a software implementation of an algorithm for approximate calculation of a Cauchy-type integral in the Wolfram Mathematica system, and provides an example. The integration of this implementation with a C# application is considered.*

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПАРАМЕТРОВ ЦИКЛИЧЕСКОГО КОДА НА ЭФФЕКТИВНОСТЬ ИТЕРАТИВНОГО ДЕКОДИРОВАНИЯ В КОСМИЧЕСКИХ КАНАЛАХ СВЯЗИ

Ключевые слова: *турбокоды, циклический избыточный код, ранняя остановка декодирования, космическая связь, энергоэффективность, CCSDS, итеративное декодирование.*

В статье рассматривается метод повышения энергоэффективности декодирования турбокодов стандарта CCSDS за счёт использования циклического избыточного кода в качестве критерия ранней остановки итеративного процесса. Кроме того, в статье разработана имитационная модель в среде MATLAB/Simulink для оценки эффективности предложенного алгоритма. Экспериментально подтверждено, что алгоритм ранней остановки на основе CRC-16-CCITT не ухудшает корректирующую способность турбокода при значительном снижении вычислительных затрат.

Введение. Освоение космического пространства невозможно представить без надежных способов передачи информации между космическими и наземными объектами. Сегодня успешность любых проектов зависит от качества и надежности канала связи. Однако современные космические системы сталкиваются с рядом серьёзных проблем, связанных с передачей данных на большие расстояния в условиях нестабильных каналов связи. К ним относятся значительные задержки сигнала, вызванные огромными дистанциями между космическими аппаратами и наземными станциями, а также влияние шумов и помех. Эти факторы приводят к искажению передаваемой информации, что требует применения эффективных методов коррекции ошибок для обеспечения надёжной связи.

Традиционные методы помехоустойчивого кодирования, такие как свёрточные коды, коды Рида-Соломона и турбокоды, широко используются в космических системах связи и рекомендованы стандартами CCSDS (Consultative Committee for Space Data Systems). Однако они обладают рядом недостатков, включая высокую вычислительную сложность декодирования, необходимость в значительных ресурсах для хранения кодовых таблиц и ограниченную адаптивность к изменяющимся условиям канала. В связи с этим

возрастает интерес к альтернативным подходам, среди которых особое место занимают методы оптимизации процесса декодирования.

Особую актуальность проблема энергоэффективности приобретает для малых космических аппаратов, где ресурсы бортовой энергии строго ограничены. Основным потребителем энергии в приёмном тракте является блок канального декодирования, особенно при использовании итеративных алгоритмов.

В данной работе рассматривается метод повышения энергоэффективности декодирования турбокодов за счёт использования циклического избыточного кода (CRC) в качестве критерия ранней остановки итеративного процесса.

1. Принцип работы турбокодов и циклически избыточных кодов. Турбокоды, предложенные К. Берру и А. Главье в 1993 году, представляют собой класс каскадных помехоустойчивых кодов, способных достигать производительности, близкой к пределу Шеннона. В современных системах космической связи они рекомендованы стандартами CCSDS (Consultative Committee for Space Data Systems) как основной метод канального кодирования для телеметрических каналов.

Турбокод состоит из каскада параллельно соединённых систематических кодов, называемых компонентными кодами. В качестве компонентных кодов могут использоваться свёрточные коды, код Хэмминга, код Рида-Соломона, код Боуза-Чоудхури-Хоквингема и другие.

В зависимости от выбора компонентного кода выделяют:

- **свёрточные турбокоды**, где компонентными кодами являются рекурсивные систематические свёрточные коды (RSC);
- **блоковые коды-произведения**, где компонентными кодами являются блочные коды (например, Хэмминга или BCH).

Кодирование осуществляется следующим образом: исходная информационная последовательность подаётся на первый компонентный кодер, а также на вход второго кодера через блок межсимвольной перестройки (интерливер). Это позволяет распределить ошибки по разным кодовым словам. Ключевой особенностью турбокодов является зависимость качества декодирования от числа итераций. В стандартных реализациях число итераций фиксируется заранее для гарантии работы в худших условиях канала. Таким образом, турбокоды обеспечивают высокую помехоустойчивость за счёт итеративного обмена информацией между компонентными декодерами. Однако фиксированное число итераций приводит к избыточным вычислениям при благоприятных условиях канала, что обуславливает необходимость разработки методов адаптивной остановки декодирования.

Циклический избыточный код (Cyclic Redundancy Check, CRC) – это алгоритм нахождения контрольной суммы, предназначенный для проверки целостности данных. В отличие от турбокодов, CRC не исправляет ошибки, а только обнаруживает их.

CRC основан на операции деления полиномов в поле Галуа $GF(2^n)$. Данные представляются как многочлен $M(x)$, который делится на фиксированный многочлен-генератор $G(x)$. Остаток от деления $R(x)$ добавляется к данным. CRC широко используется в стандартах CCSDS для формирования кадров телеметрии. Обычно контрольная сумма вычисляется для всего кадра данных и передаётся в конце пакета. В классической схеме проверка CRC происходит только после полного завершения декодирования канала. Если декодер ошибся, CRC это обнаружит, и кадр будет запрошен повторно или отброшен.

Высокая помехоустойчивость турбокодов обеспечивается за счёт итеративного процесса декодирования. Однако этот механизм имеет обратную сторону: вычислительная сложность декодера линейно зависит от числа итераций. Для гарантии надёжности в худших условиях канала количество итераций фиксируется на максимальном уровне, обычно от 6 до 10. В реальных условиях космической связи отношение «сигнал-шум» часто меняется. При благоприятных условиях (высокое SNR) сходимость алгоритма достигается за 2–4 итерации. Продолжение декодирования до максимального значения приводит к выполнению избыточных вычислений, что напрямую увеличивает энергопотребление бортовой аппаратуры без улучшения качества связи. Для устранения этой избыточности существуют методы ранней остановки, которые позволяют прервать процесс декодирования досрочно, если достигнуто надёжное решение.

Основные классы методов включают:

- 1) **методы на основе синдрома** – проверка равенства нулю синдрома декодированного слова;
- 2) **методы на основе достоверности (LLR Threshold)** – анализ абсолютных значений логарифмических отношений правдоподобия;
- 3) **Методы на основе контрольных сумм (CRC-aided)** – проверка целостности данных с помощью циклического кода.

Несмотря на наличие методов, задача выбора критерия остановки для космических систем остаётся открытой. Критерий должен обеспечивать баланс между тремя противоречивыми требованиями: минимальная вычислительная сложность реализации, высокая надёжность обнаружения сходимости и совместимость со стандартами космической связи (CCSDS).

2. Постановка задачи. Существующие методы ранней остановки, рассмотренные выше имеют ряд ограничений, затрудняющих их

применение в условиях ограниченных ресурсов в космических системах. Методы на основе синдрома, требуют доступа к проверочной матрице кода и выполнения операций матричного умножения на каждой итерации. Для турбокодов большой длины это создаёт значительную нагрузку на вычислительное ядро процессора, нивелируя экономию от ранней остановки. Методы на основе порога LLR требуют эмпирической настройки порогового значения для каждого конкретного канала связи. В условиях космоса, где параметры канала могут меняться, фиксированный порог может приводить либо к преждевременной остановке, либо к отсутствию или неэкономичному использованию ресурсов.

Таким образом, необходима разработка метода, который не требует сложной матричной арифметики, не зависит от тонкой настройки порогов и использует уже существующую инфраструктуру защиты данных.

В настоящей работе предлагается использовать циклический избыточный код в качестве критерия ранней остановки. Этот метод был выбран так как согласно рекомендациям CCSDS, кадры телеметрии уже защищаются кодами CRC и использование существующих битов контроля не требует увеличения избыточности канала связи. Алгоритм вычисления CRC реализуется через операции сдвига и логической суммы, что значительно проще, чем одна итерация турбодекодера или вычисление синдрома.

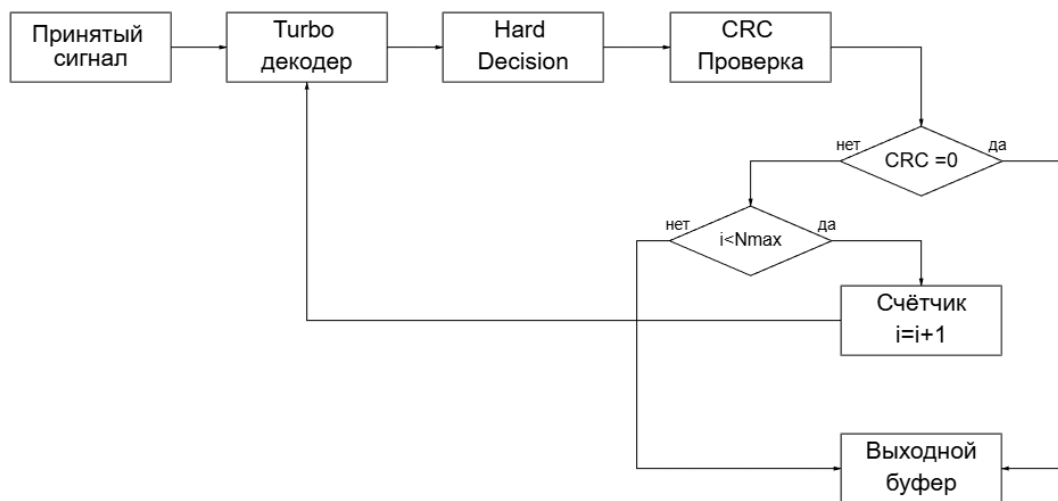


Рис. 1. Блок-схема процесса декодирования полученной последовательности

Для решения проблемы энергоэффективности декодирования в работе разработан алгоритм итеративного декодирования турбокодов с интегрированной проверкой CRC на каждом шаге итерации. Оценено влияние параметра SNR на среднее число итераций и вероятность

битовой ошибки (BER). Количественно оценена потенциальная экономия вычислительных ресурсов без деградации надёжности передачи данных.

Математически опишем алгоритм. Пусть U – информационный блок длины K бит. На передающей стороне формируется кадр $V = \{U, CRC(U)\}$, который кодируется турбокодером и передаётся по каналу AWGN. На приёмной стороне поступает вектор Y . Предлагаемый алгоритм заменяет фиксированное число итераций N_{max} на переменное N_{stop} . Процесс декодирования показан на блок-схеме (рис. 1).

3. Экспериментальные исследования эффективности алгоритма ранней остановки декодирования турбокодов. Для оценки эффективности алгоритма ранней остановки декодирования турбокодов на основе проверки циклическим избыточным кодом (CRC) была разработана имитационная модель в среде MATLAB/Simulink.

Модель воспроизводит полный цикл передачи данных и включает последовательную цепь обработки сигналов. Источником информации служит генератор случайных битов. Сформированный информационный поток поступает на блок кодирования, где к кадру добавляется 16-битная контрольная сумма для последующего обнаружения ошибок. Затем данные кодируются турбокодером стандарта с использованием свёрточных кодов и подвергаются фазовой манипуляции в модуляторе. Передача осуществляется по каналу связи с аддитивным белым гауссовским шумом (AWGN). На приёмной стороне сигнал демодулируется, после чего поступает на вход турбодекодера. Верификация декодированных данных выполняется блоком проверки CRC, результат которой формирует сигнал управления для блока While Iterator, регулирующего число итераций декодирования. Выходные сигналы модели сохраняются в рабочем пространстве.

Алгоритм ранней остановки функционирует в цикле итеративного декодирования. После каждой итерации работы турбодекодера вычисляется циклическая контрольная сумма для полученных данных. В случае успешного прохождения проверки CRC декодирование прекращается досрочно, что позволяет снизить вычислительные затраты без потери информации. Если проверка не пройдена, выполняется следующая итерация вплоть до достижения максимального установленного значения. Таким образом, среднее число итераций адаптируется к текущим условиям канала связи, определяемым отношением сигнал / шум, обеспечивая оптимальное распределение вычислительных ресурсов. Ключевым преимуществом данного подхода является использование уже существующего в системе механизма CRC для обнаружения ошибок, что не требует дополнительных аппаратных затрат. Алгоритм гарантирует отсутствие ухудшения корректирующей

способности кода, так как ранняя остановка происходит только при подтверждённой корректности декодированных данных.

Оценивалось два ключевых параметра вероятность битовой ошибки (BER) и среднее число итераций декодирования, отражающее вычислительную сложность алгоритма.

Вероятность битовой ошибки рассчитывалась по формуле:

$$BER = \frac{N_{errors}}{N_{total}}, \quad (1)$$

где N_{errors} – количество ошибочно принятых битов, N_{total} – общее количество переданных битов.

Среднее число итераций вычислялось как среднее арифметическое по всем кадрам:

$$I_{avg} = \frac{\sum_{k=1}^{N_{frames}} I_k}{N_{frames}}, \quad (2)$$

где I_k – число итераций для k-го кадра, N_{frames} – общее число кадров.

Экономия вычислительных ресурсов определялась по формуле:

$$\text{Экономия, \%} = \left(1 - \frac{I_{avg}}{I_{max}}\right) \times 100\%, \quad (3)$$

где I_{max} – максимальное число итераций (8).

В таблице 1 представлены результаты измерения BER для системы с ранней остановкой и без неё.

Таблица 1

Зависимость BER от SNR

SNR, дБ	BER (без остановки)	BER (с ранней остановкой)	Отклонение, %
0	$8,42 \times 10^{-2}$	$8,45 \times 10^{-2}$	0,36
1	$5,23 \times 10^{-2}$	$5,26 \times 10^{-2}$	0,57
2	$2,87 \times 10^{-2}$	$2,89 \times 10^{-2}$	0,70
3	$5,42 \times 10^{-3}$	$5,45 \times 10^{-3}$	0,55
4	$5,23 \times 10^{-4}$	$5,25 \times 10^{-4}$	0,38
5	$2,34 \times 10^{-5}$	$2,35 \times 10^{-5}$	0,43
6	$5,42 \times 10^{-7}$	$5,43 \times 10^{-7}$	0,18

Анализ данных таблицы 1 показывает, что, если ввести в систему раннюю остановку, это критически не повлияет на надёжность всей системы, а значит, CRC не ухудшает корректирующую способность турбокода

В таблице 2 представлена оценка экономии вычислительных ресурсов за счёт ранней остановки.

Таблица 2

Экономия вычислительных ресурсов

SNR, дБ	Среднее итераций	Экономия, %
0	7,82	2,25
1	7,45	6,88
2	6,87	14,13
3	5,23	34,63
4	3,45	56,88
5	2,23	72,13
6	1,56	80,50

При SNR < 2 дБ экономия незначительна (< 15%). При SNR = 3–4 дБ экономия составляет 35–57%. При SNR > 5 дБ экономия превышает 70%. Алгоритм обеспечивает значительную экономию в рабочем диапазоне SNR.

Заключение. Стоит отметить, что алгоритм ранней остановки на основе CRC-16-CCITT не ухудшает корректирующую способность турбокода: отклонение BER составляет менее 1% во всём диапазоне SNR. При SNR = 3–5 дБ экономия числа итераций – 35–72%, что представляет значительный интерес для практического применения. Снижение числа итераций приводит к уменьшению энергопотребления декодера на 35–80%, что критично для автономных устройств и спутниковых систем связи.

Предложенный алгоритм превосходит известные методы по сочетанию экономии вычислений и сохранения корректирующей способности. Алгоритм может быть реализован в реальных системах связи без существенных модификаций аппаратного обеспечения, так как CRC обычно уже присутствует в системах для обнаружения ошибок. Для максимального эффекта алгоритм ранней остановки рекомендуется применять в системах, работающих при SNR > 3 дБ. При более низких SNR экономия незначительна и целесообразно использовать фиксированное число итераций.

Литература

1. Wicker S.B. Error Control Systems for Digital Communication and Storage. Prentice Hall, 1995.
2. Richardson T., Urbanke R. Modern Coding Theory. Cambridge University Press, 2008.
3. Сергеев В.С., Баринов В.В. Сжатие данных, речи, звука и изображений в телекоммуникационных системах. М.: РадиоСофт, 2009. 360 с.
4. Майоров А.А., Седякин В.П. Общая и теоретическая информатика: учебное пособие. М.: МИИ-ГАиК, 2017. 128 с.
5. Беспалова Н.М., Сологубова Л.А., Байбекова Ф.Н. Обзор основных алгоритмов сжатия данных // Информационная безопасность – актуальная проблема современности. Совершенствование образовательных технологий подготовки

специалистов в области информационной безопасности. 2019. № 1 (10). С. 145–148.

6. Дадаев Ю.Г. Теория арифметических кодов. М.: Радио и связь, 1981. 272 с.

7. Склад Б. Цифровая связь. Теоретические основы и практическое применение: пер. с англ. 2-е изд. М.: Издательский дом «Вильямс», 2003. 1104 с.

V.V. Kudinov

Russian Technological University MIREA

INVESTIGATION OF THE EFFECT OF CYCLIC CODE PARAMETERS ON THE EFFECTIVENESS OF ITERATIVE DECODING IN SPACE COMMUNICATION CHANNELS

Keywords: *Turbo codes, cyclic redundancy code, early decoding stop, space communication, energy efficiency, CCSDS, iterative decoding.*

Abstract. *This paper discusses a method for improving the energy efficiency of turbo code decoding in the CCSDS standard by using a cyclic redundancy code as a criterion for early stopping the iterative process. Additionally, the paper develops a simulation model in the MATLAB/Simulink environment to evaluate the effectiveness of the proposed algorithm. It has been experimentally confirmed that the early stop algorithm based on CRC-16-CCITT does not degrade the turbo code's error correction capability while significantly reducing computational costs.*

Н.С. Миронов

Национальный исследовательский университет «МИЭТ»

УДК 004.042

РАЗРАБОТКА МОДЕЛИ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ОШИБОК ПРИ СБОРКЕ ПЕЧАТНЫХ ПЛАТ

Ключевые слова: *модель глубокого обучения, печатные платы, диагностика, сверточные нейронные сети, дефекты, извлечение признаков, классификация, сегментация, прогнозирование, энтропия.*

В статье представлена разработка модели глубокого обучения, специально предназначенной для выявления ошибок при сборке электронных компонентов на печатной плате. Предложенная модель сверточного глубокого обучения состоит из трех этапов: сегментации, извлечения признаков и классификации. Сегментация основана на модели Mask R-CNN для обнаружения объектов на изображениях. На этапе извлечения признаков используются сверточные нейронные сети

и подход, основанный на принципах человеческого зрения: сначала выявляются грубые структуры, затем средние детали и, наконец, мелкие. Этап классификации выполняется с помощью полносвязных нейронных сетей. После обучения и тестирования на этапе сегментации была достигнута средняя точность 100 %. Что касается извлечения и классификации атрибутов, то точность составила 100 %, а количество ошибок было равно нулю как при обучении, так и при тестировании, что подтверждает высокую эффективность этих этапов по отдельности.

К поверхностным дефектам печатных плат относятся, в частности, короткие замыкания, обрывы цепей и т.д. Дефекты пайки связаны, в частности, с недостаточным или избыточным количеством припоя на выводах компонентов или с короткими замыканиями, вызванными неправильной техникой пайки [1].

Ошибки при сборке связаны с неправильной установкой некоторых электронных компонентов на печатную плату, например с отсутствием элементов, их неправильным расположением или нарушением полярности [2–3]. Из-за сложности этой проблемы и с целью ее решения несколько авторов разработали методики, ориентированные на распознавание только тех устройств, которые присутствуют на этих платах. В таблице 1 представлено сравнение четырех наиболее конкурентоспособных работ 2024 и 2025 годов. Во всех работах в качестве критерия сравнения используется точность.

Несмотря на то, что предыдущие методы показали конкурентоспособные результаты, они часто требовали использования контролируемой среды и с трудом адаптировались к различным типам дефектов. В этом исследовании представлена модель глубокого обучения, которая сочетает в себе точную сегментацию с иерархическим извлечением признаков, что позволяет проводить надежный контроль печатных плат в режиме реального времени. Существующие системы контроля печатных плат редко обеспечивают идеальную точность классификации в режиме реального времени при использовании собственных наборов данных [5]. Проектирование разработанной модели глубокого обучения состоит из трех этапов: сегментации, выделения атрибутов и классификации.

В первом используется модель Mask R-CNN для обнаружения, сегментации и извлечения печатных плат на цифровых изображениях в виде объектов.

Вторая извлекает соответствующую информацию для выполнения процесса классификации и состоит из трех сверточных нейронных сетей, работающих по схеме «Грубый-средний-тонкий». На последнем этапе извлеченные атрибуты анализируются с помощью полносвязной

искусственной нейронной сети для классификации досок по типу дефектов.

Таблица 1

**Сравнение методов обнаружения дефектов печатных плат
(современные методы)**

Название	Основные характеристики	Точность (%)
MAS-YOLO: упрощенный алгоритм обнаружения дефектов печатных плат на основе усовершенствованной модели YOLOv12 [4]	Улучшение YOLOv12. Медианно-усиленная CSA (пространственное и канальное внимание). Адаптивная иерархическая сеть признаков	93.2
Легкий метод обнаружения дефектов печатных плат на основе SCF-YOLO	Улучшенная версия YOLOv8. MobileNet в качестве средства извлечения признаков. Модуль SCF + функция объединения обучаемых весов. Функция потерь CIOU + GIOU	92.5
Алгоритм обнаружения дефектов печатных плат на основе CDI-YOLO [5]	Улучшение YOLOv7-tiny. Модуль внимания CA. Замена слоя на DSCONV. Функция потерь Inner-CIOU	97.1
Обнаружение дефектов при сборке печатных плат на основе YOLOv5 [6]	Улучшенный набор данных YOLOv5. PCBA-DET (4000 изображений, 8 дефектов, 18 100 областей интереса). Извлечение признаков с помощью RepLKNet + шейп-сет с SPPCSPC. Замена функции потерь CIOU на SIOU	96.4

На рисунке 1 представлена предлагаемая модель глубокого обучения.

Этап сегментации преимущественно основан на модели Mask R-CNN (свёрточная нейронная сеть на основе маскирования областей) версии 2.1, которая после обучения способна генерировать точные маски для каждого объекта на изображении. Ниже представлены основные слои этой глубокой сети.

Извлечение признаков. Для выполнения задачи в маске R-CNN используется сверточная сеть ResNet-101. Кроме того, в ней есть блоки «пирамидальной сети признаков» (Feature Pyramid Network, FPN), которые улучшают обнаружение объектов на изображениях разных размеров.

Сеть предложений по областям интереса (RPN). Данный тип сети используется для генерации предложений по областям интереса (ROI) объектов на изображении. Для этого RPN использует несколько ограничивающих рамок с разным соотношением сторон и размерами.

Сеть выравнивания областей интереса (RoI Align). По сути, она преобразует области интереса, сгенерированные RPN, в области фиксированного размера, которые могут быть легко обработаны последующими слоями сети. Это вариант сети объединения областей интереса, используемой в Faster R-CNN, которая повышает точность сегментации. Здесь используется билинейная интерполяция для вычисления точных значений атрибутов в каждой области интереса, что обеспечивает точное пространственное выравнивание.

Слой прогнозирования. В маске R-CNN используются три типа слоев прогнозирования: Bounding Box Top, Classification Top и Mask Top. Первый слой предсказывает ограничивающие рамки для каждой предложенной области интереса, второй классифицирует эти области по одному из возможных классов объектов, а третий генерирует бинарную маску на уровне пикселей для каждого элемента, обнаруженного на изображении. Эта маска дорабатывается для каждого объекта, что позволяет добиться точной сегментации и различать элементы одного класса на одном изображении.

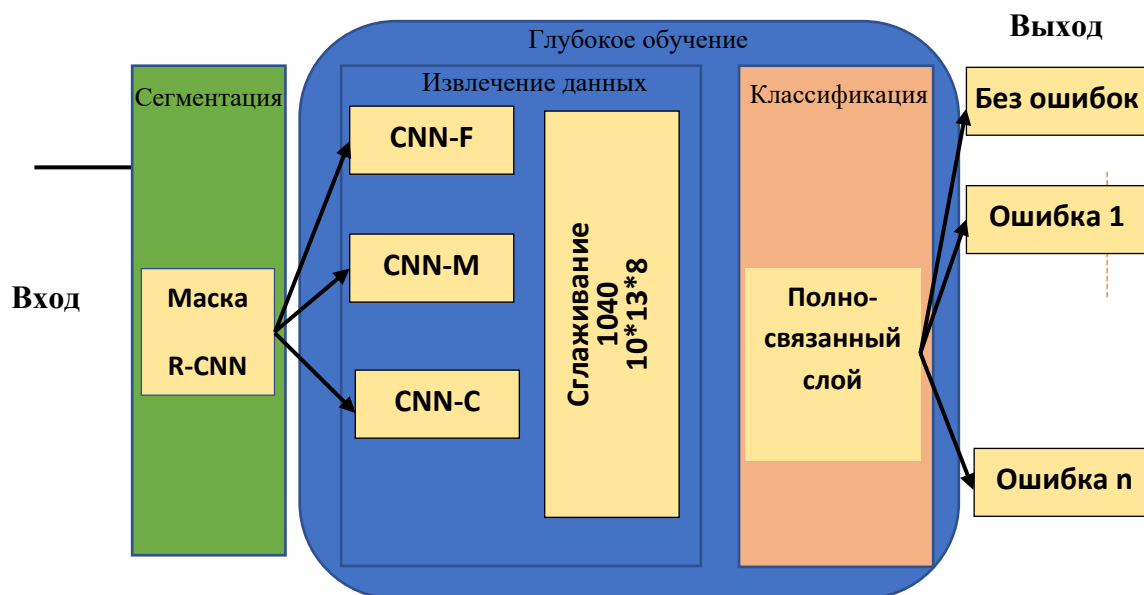


Рис. 1. Предлагаемая модель глубокого обучения

Исходя из вышесказанного, после того как Mask R-CNN сгенерирует маску, соответствующую обнаруженной печатной плате, извлекается только эта плата. Таким образом, на этапе сегментации используется дополнительный блок, который выполняет эту задачу с

помощью функции И между входным изображением и полученной маской.

Этап извлечения признаков в основном состоит из трех сверточных нейронных сетей по схеме «крупный-средний-тонкий». Этот подход основан на зрении человека: сначала воспринимаются грубые структуры, затем средние и, наконец, мелкие детали. Эти три сети подключены параллельно и получили названия: CNN-F, CNN-M и CNN-S. Технические характеристики каждой сети описаны ниже. Для всех свёрточных слоёв параметр заполнения был настроен на «нулевое заполнение».

CNN-F. Эта сеть состоит из шести двумерных свёрточных слоёв для извлечения признаков. Первый слой, также называемый входным, получает изображение 682×853 пикселей в формате RGB и состоит из 32 фильтров. Второй и третий слои содержат по 64 фильтра, а четвертый, пятый и шестой слои – по 32, 16 и 8 фильтров соответственно. Стоит отметить, что размер фильтров в первом слое составляет 7×7 , а для остальных – 5×5 . Кроме того, в конце каждого блока извлечения признаков использовалась функция активации Leaky-ReLU с альфа-коэффициентом 0,2 и максимальным пулом 2×2 (рис. 2).

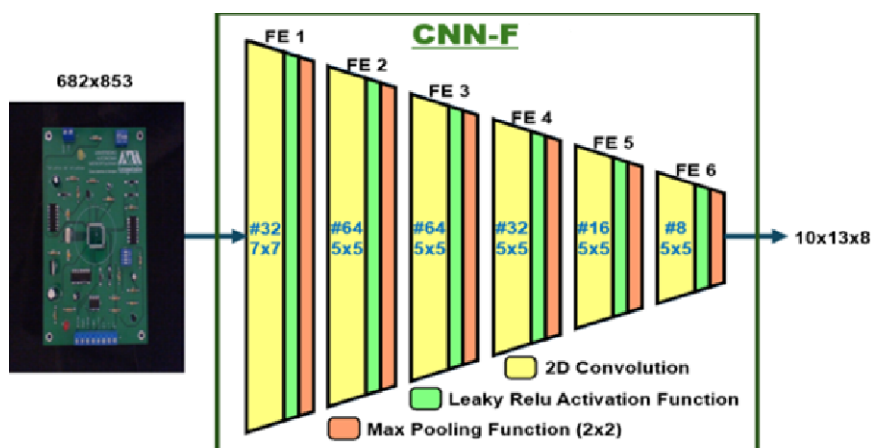


Рис. 2. Архитектура сети CNN-F

CNN-M. Эта сеть состоит из четырех двумерных свёрточных слоев для извлечения признаков. Первый слой получает изображение 682×853 пикселей в формате RGB и состоит из 64 фильтров. Второй слой содержит 32 фильтра, а третий и четвертый – 16 и 8 фильтров соответственно. Стоит отметить, что размер фильтров первого слоя составляет 7×7 , а для остальных – 5×5 . Кроме того, в конце каждого блока извлечения признаков использовалась функция активации Leaky ReLU с альфа-коэффициентом 0,2 и максимальным пулом 2×2 . Наконец, чтобы размерность на выходе соответствовала

размерности тонкой сети, используется шаг 2×2 . Для третьего и четвертого слоев свертки использовался сдвиг на два пикселя (рис. 3).

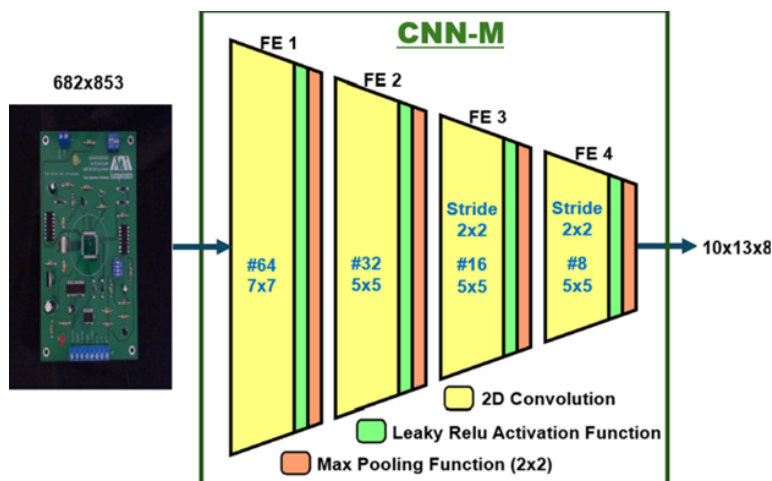


Рис. 3. Архитектура сети CNN-M

CNN-C. Эта сеть состоит из трех двумерных сверточных слоев для извлечения признаков. Первый слой получает изображение 682×853 пикселей в формате RGB и состоит из 64 фильтров 7×7 . Кроме того, в конце каждого блока извлечения признаков использовалась функция активации Leaky ReLU с альфа-коэффициентом 0,2 и максимальным пулом 2×2 .

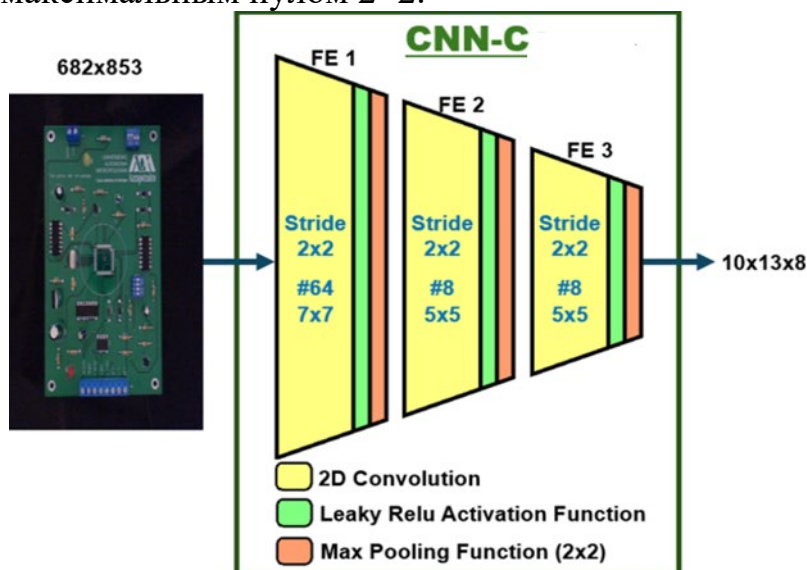


Рис. 4. Архитектура сети CNN-C.

Чтобы получить общий вектор признаков, необходимо объединить признаки, извлеченные каждой из сетей (CNN-F, CNN-M и CNN-C). В соответствии с парадигмой зрительной системы человека мелкие детали распознаются сетью CNN-F, поэтому ей придается больший вес. Затем следует объединение мелких деталей со средними структурами, распознаваемыми сетью CNN-M. Сеть CNN-C предоставляет

информацию о крупных структурах и макрокомпонентах. Таким образом, веса расставляются следующим образом: CNN-F -0,5; CNN-M-0,3 и CNN-C – 0,2.

Общая функция потерь, соответствующая модели глубокого обучения, разработанной в этой статье, описывается уравнением (1).

$$\Omega_{\text{общ}}(\varsigma_1, \varsigma_2, \varsigma_3, \mu) = \Omega_{\text{извл}}(\varsigma_1, \varsigma_2, \varsigma_3) + \Omega_{\text{клас}}(\mu), \quad (1)$$

где $\Omega_{\text{извл}}(\varsigma_1, \varsigma_2, \varsigma_3)$ – функция потерь, соответствующая задаче извлечения признаков; $\Omega_{\text{клас}}(\mu)$ – функция потерь, соответствующая задаче классификации; $\varsigma_1, \varsigma_2, \varsigma_3$ – набор параметров, соответствующий задачам CNN-F, CNN-M, CNN-C.

Поскольку общая модель глубокого обучения состоит из этапов извлечения признаков и классификации, эти этапы разбиты на части в соответствии с наборами параметров. Этап извлечения признаков состоит из трех ветвей: тонкой (параметр ς_1), средний (параметр ς_2) и грубый (параметр ς_3). Параметры этапа классификации выражаются через параметр μ . Таким образом, оптимизация выполняется для шести типов плат.

Функцию полных потерь $\Omega_{\text{общ}}$ можно переписать в терминах энтропии, как показано в уравнении (2).

$$\begin{aligned} \Omega_{\text{общ}}(\varsigma_1, \varsigma_2, \varsigma_3, \mu) = \\ = \sum_{j=1}^6 -\log[\hat{p}(C_j / \varsigma_1)] - \log[\hat{p}(C_j / \varsigma_2)] - \log[\hat{p}(C_j / \varsigma_3) - \log[\hat{p}(C_j / \mu)]] \end{aligned} \quad (2)$$

где C_j – маркер правильности собранной платы; $\hat{p}(C_j / \varsigma_1)$ – условная вероятность для платы, зависящей от набора параметров, соответствующих задаче тонкой настройки CNN; $\hat{p}(C_j / \varsigma_2)$ – условная вероятность для платы, зависящей от набора параметров, соответствующих задаче CNN-M; $\hat{p}(C_j / \varsigma_3)$ – условная вероятность для платы, зависящей от набора параметров, соответствующих задаче CNN-C; $\hat{p}(C_j / \mu)$ – условная вероятность для платы, зависящей от набора параметров, соответствующих задаче классификации.

Оптимизируя уравнение (2) с помощью градиентного спуска, имеем:

$$\nabla_{(\varsigma_1, \varsigma_2, \varsigma_3, \mu)} \Omega_{\text{общ}} \left[\frac{1}{M_B} \sum_X \log \Omega_{\text{общ}} \right]. \quad (3)$$

Наконец, уравнение (3) используется при обучении для поиска минимальной ошибки в уравнении (2), которая должна быть равна нулю.

Функция категориальной кросс-энтропийной потери используется при обучении и оценке для определения ошибки модели.

Обучение разработанной модели глубокого обучения было разделено на два этапа. Сначала была обучена модель Mask R-CNN, которая используется на этапе сегментации, а затем – этапы извлечения атрибутов и классификации.

Для обучения Mask R-CNN модули, составляющие эту модель глубокого обучения, были сначала загружены с сайта github.com. Это было сделано из официального репозитория разработчика. В процессе обучения использовался оптимизатор стохастического градиентного спуска (SGD) со скоростью обучения 0,001 и значением момента 0,9, а также с коэффициентом затухания весов, равным 0,0001. Модель Mask R-CNN обучалась с использованием SGD с импульсом в качестве оптимизатора, поскольку он более стабилен и менее подвержен переобучению, чем Adam. Гиперпараметры – скорость обучения, импульс и коэффициент затухания – были скорректированы до значений, указанных в статье, чтобы обеспечить сходимость за минимальное количество эпох и наилучшие результаты тестирования.

Используемый набор данных состоял из 173 изображений с соответствующими файлами в формате JSON, которые, как упоминалось выше, содержат координаты печатной платы на каждом изображении. С использованием этих данных и параметров модель Mask R-CNN была обучена на слоях прогнозирования в течение пяти эпох с шагом 131.

Для обучения были созданы этапы извлечения признаков и классификации с использованием функции категориальной кросс-энтропийной потери и оптимизатора Adam (размера пакета = 128, количества эпох = 14 и периода выборки = 50). На этапах извлечения признаков и классификации для достижения более быстрой сходимости использовался оптимизатор Adam, поскольку база данных относительно небольшая. Значение скорости обучения на этих этапах было выбрано на основе опыта и положительных результатов, полученных при обучении сетей с характеристиками, аналогичными тем, что использовались в этом исследовании.

Затем глубокая нейронная сеть обучалась в течение 15 эпох с размером пакета 10, используя 80% описанной выше базы данных. Таким образом, из 2376 изображений для этой задачи было использовано 1900. По завершении обучения ошибка была сведена к нулю, а точность составила 100 %. На графиках на рисунке 5 показано изменение показателей по эпохам. Из этих графиков видно, что максимальная производительность была достигнута на седьмой эпохе.

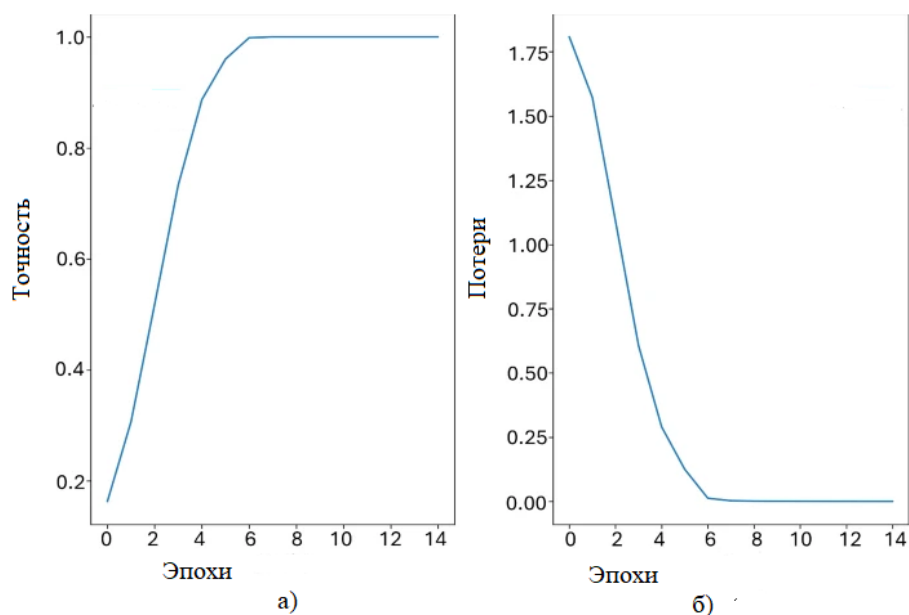


Рис. 5. Показатели точности и потерь в зависимости от эпох обучения

Литература

1. Yang J., Li S., Wang Z., Dong H., Wang J., Tang S. (2020). Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges.
2. Janiesch C., Zschech P., Heinrich K. (2021). Machine learning and deep learning. *Electronic Markets*. 31(3). P. 685–695. URL: <https://doi.org/10.1007/s12525-021-00475-2>.
3. Kulkarni R., Kulkarni S., Dabhane S., Lele N., Paswan R.S. (2019). An automated computer vision based system for bottle cap fitting inspection // 2019 Twelfth international conference on contemporary computing (IC3) (p. 1–5). URL: <https://doi.org/10.1109/IC3.2019.8844942>.
4. Yin X., Zhao Z., Weng L. (2025). MAS-YOLO: A lightweight detection algorithm for PCB defect detection based on improved YOLOv12. *Applied Sciences*, 15(11), 6238. URL: <https://doi.org/10.3390/app15116238>.
5. Ficzer M., Meszaros L.A., Kallai-Szabo N., Kovacs A., Antal I., Nagy Z.K., Galata D.L. (2022). Real-time coating thickness measurement and defect recognition of film coated tablets with machine vision and deep learning. *International Journal of Pharmaceutics*. 623. Article 121957. URL: <https://doi.org/10.1016/j.ijpharm.2022.121957>.
6. Lian J., Wang L., Liu T., Ding X., Yu Z. (2021). Automatic visual inspection for printed circuit board via novel mask R-CNN in smart city applications. *Sustainable Energy Technologies and Assessments*, 44, Article 101032. URL: <https://doi.org/10.1016/j.seta.2021.101032>.
7. Xiao G., Hou S., Zhou H. (2024). PCB defect detection algorithm based on CDI-YOLO. *Scientific Reports*, 14, 7351. URL: <https://doi.org/10.1038/s41598-024-57491-3>.

N.S. Mironov

National Research University of Electronic Technology

DEVELOPMENT OF A DEEP LEARNING MODEL FOR DETECTING ERRORS DURING PCB ASSEMBLY

Keywords: *deep learning model, printed circuit boards, diagnostics, convolutional neural networks, defects, feature extraction, classification, segmentation, prediction, entropy*

Abstract. *This article presents the development of a deep learning model specifically designed to detect errors in the assembly of electronic components on a printed circuit board. The proposed convolutional deep learning model consists of three stages: segmentation, feature extraction, and classification. Segmentation is based on the Mask R-CNN model for detecting objects in images. At the stage of feature extraction, convolutional neural networks and an approach based on the principles of human vision are used: first, coarse structures are identified, then medium details, and finally small ones. The classification stage is performed using fully connected neural networks. After training and testing at the segmentation stage, an average accuracy of 100% was achieved. As for the extraction and classification of attributes, the accuracy was 100%, and the number of errors was zero in both training and testing, which confirms the high efficiency of these stages individually.*

Н.С. Миронов, Ю.Ю. Юдахин

Национальный исследовательский университет «МИЭТ»

УДК 004.042

РАЗРАБОТКА МЕТОДА СЕГМЕНТАЦИИ ДЕФЕКТОВ ПАЙКИ ПЕЧАТНЫХ ПЛАТ

Ключевые слова: *сегментация изображений, дефекты пайки, нейронная сеть UNet, канальное и пространственное внимание, гибридный модуль, точность.*

В силу ряда проблем точная сегментация дефектов пайки является сложной задачей. Чтобы решить эту проблему, для сегментации дефектов пайки на печатных платах (ПП) предлагается метод улучшения структуры сети кодировщика-декодировщика UNet. Чтобы расширить возможности кодировщика по извлечению признаков и сосредоточиться на более глубоких признаках, в качестве кодировщика сети используется VGG16. Разработан гибридный модуль внимания DHAM, который сочетает канальное внимание и динамическое пространственное внимание. DHAM подавляет фоновые помехи и усиливает отклики в областях с дефектами, повышая селективность признаков. В декодере на основе GSConv применён модуль RGSM (Residual GSConv-based Merge), обеспечивающий более эффективное

объединение многомасштабных признаков через комбинацию легковесных свёрток, канального перемешивания и резидуальной связи, что повышает точность сегментации. Результаты экспериментов показали, что предлагаемый метод может эффективно повысить точность сегментации дефектов пайки печатных плат, достигнув $mIoU$ 81,7% и mPA 87,3%.

Обнаружение дефектов печатных плат стало важнейшей технологией в современной электронной промышленности. В настоящее время основные подходы к обнаружению дефектов основаны на машинном обучении, которое преимущественно делится на традиционное машинное обучение и глубокое обучение. Как показано в таблице 1, традиционный метод обнаружения дефектов с помощью машинного обучения обычно предполагает извлечение таких характеристик, как размер, цвет и геометрическая форма целевого изображения, а затем ввод этих характеристик в классификатор для классификации.

Однако эти традиционные методы машинного обучения требуют много времени и усилий для извлечения характеристик в случае сложных образцов, а также разработки соответствующих алгоритмов извлечения характеристик для различных задач обнаружения.

Таблица 1

Классификация методов обнаружения дефектов, их преимущества и недостатки

Тип метода	Имя	Преимущество	Недостаток
Традиционные методы обнаружения машинного обучения	Метод обнаружения классификатора на основе характеристик изображения	Метод простой и незамысловатый	Занимает много времени и не очень точен
Алгоритмы обнаружения на основе глубокого обучения	Faster R-CNN, Mask R-CNN и т. д.	Дефекты цели могут быть обнаружены автоматически	Низкая точность классификации и обнаружения дефектов в нескольких изделиях
	Предлагаемый метод	Высокая точность классификации и обнаружения бракованных изделий с несколькими целями	Значительный объем вычислений для больших моделей

Алгоритмы обнаружения на основе глубокого обучения для сквозного анализа могут стать хорошим решением для устранения недостатков традиционных подходов к машинному обучению, но для обучения моделей глубокого обучения требуется большое количество наборов данных с ошибками [1; 2].

В последние несколько лет исследователи часто использовали состязательные генеративные сети для создания изображений, чтобы расширить набор данных. Все эти методы состязательной генерации требуют разделения выборки на два разных класса для конкурентного сопоставления, то есть на дефектные и недефектные образцы, но при работе с дефектами сварки печатных плат, такими как дефектные и недефектные наборы данных, хороших результатов добиться не удаётся. Диффузионную модель можно использовать в целях расширения выборки данных, поскольку для создания выборочных изображений, очень похожих на исходные, ей нужны только определённые подсказки и исходные изображения.

Алгоритмы обнаружения на основе глубокого обучения извлекают информацию о характеристиках целевых изображений путём обучения нейронных сетей, которые могут автоматически распознавать сложные и изменчивые целевые объекты [3; 4]. В работе [5] использована Faster R-CNN для обнаружения дефектов на лопастях турбин, что повысило точность обнаружения дефектов, а также подтвердило превосходство этого метода в скорости обнаружения. В работе [6] использована Mask R-CNN для обнаружения дефектов при пайке выводов резисторов на печатных платах, что позволило классифицировать и маркировать множество дефектов. Однако при наличии схожих электронных компонентов на печатной плате, таких дефектов сварки, как сложный фон, сложная форма дефекта и т.д., описанная выше сеть обнаружения не даёт хороших результатов.

Таким образом, при обнаружении дефектов пайки электронных компонентов на печатных платах из-за разнообразия целевых дефектов традиционные методы машинного обучения являются трудоёмкими и сложными, а методы обнаружения на основе глубокого обучения, такие как Faster R-CNN и Mask R-CNN, не обеспечивают высокую точность обнаружения. Поэтому необходимо усовершенствовать сеть обнаружения.

U-Net – это классическая структура свёрточной нейронной сети с симметричной структурой «кодировщик – декодировщик», которая обеспечивает передачу информации от глобального контекста к локальным деталям с помощью путей понижающей и повышающей дискретизации. В Faster R-CNN представлена сеть предложений регионов (Region Proposal Network, RPN) и сеть классификации регионов для быстрого и точного обнаружения объектов, что позволяет

оптимизировать смещение ограничительной рамки за счёт вычислительного метода. Mask R-CNN представляет собой дополнительную разветвлённую сеть, которая позволяет модели одновременно генерировать маски на уровне пикселей для каждого обнаруженного объекта, тем самым одновременно реализуя как обнаружение объектов, так и семантическую сегментацию, а также улучшая визуальный эффект обнаружения целей [7].

ROI Pooling и ROI Align – это два широко используемых метода объединения областей интереса (ROI) для извлечения признаков внутри областей интереса из карт признаков. Объединение ROI позволяет извлечь объекты путем разделения ROI на сетки фиксированного размера и объединения максимального значения каждой сетки, но этот метод страдает от потери информации. Напротив, ROI Align точно извлекает объекты, выполняя билинейную интерполяцию в ROI, что позволяет избежать потери информации и повысить точность локализации.

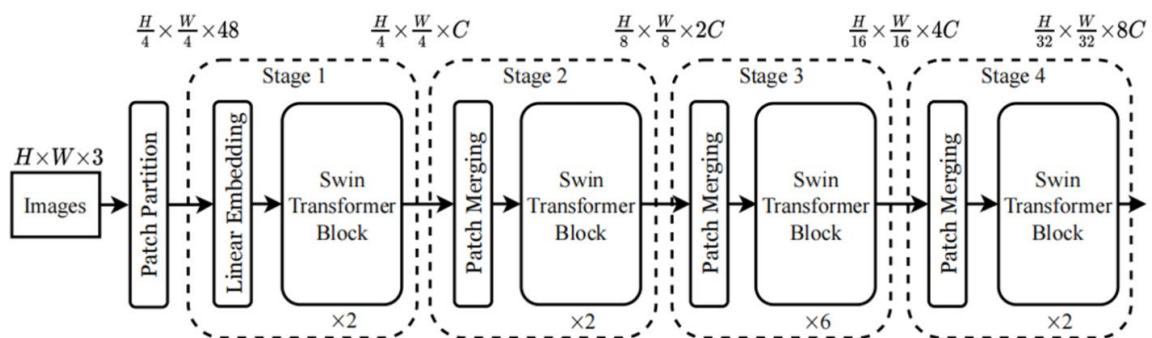
Cascade Mask R-CNN улучшает Mask R-CNN за счёт каскадного соединения нескольких перцептронов, что позволяет постепенно отфильтровывать ложные срабатывания, уменьшать несоответствия и повышать эффективность и точность обнаружения. Каскадная структура состоит из нескольких этапов. Выходные данные предыдущего этапа используются в качестве входных данных для последующего этапа, и на каждом этапе применяется дополнительный порог обнаружения объектов для фильтрации кадров-кандидатов, которые считаются положительными образцами на этом этапе. Благодаря прогрессивной фильтрации с различными пороговыми значениями можно значительно снизить частоту ложных срабатываний и утечек, поэтому каскадная структура может более надёжно справляться с задачами многоцелевого обнаружения в различных сложных сценариях.

Все вышеупомянутые сети для обнаружения объектов используют ResNet или VGG в качестве базовой сети. Эти свёрточные нейронные сети используют свёрточное ядро фиксированного размера и слой объединения, которые при выборке образцов склонны игнорировать информацию о признаках на больших расстояниях. По мере углубления свёрточного слоя информация о контурах объекта будет быстро теряться, поэтому при обнаружении дефектов пайки компонентов поверхностного монтажа на печатной плате невозможно одновременно обнаружить и распознать паяные соединения и различные компоненты [8].

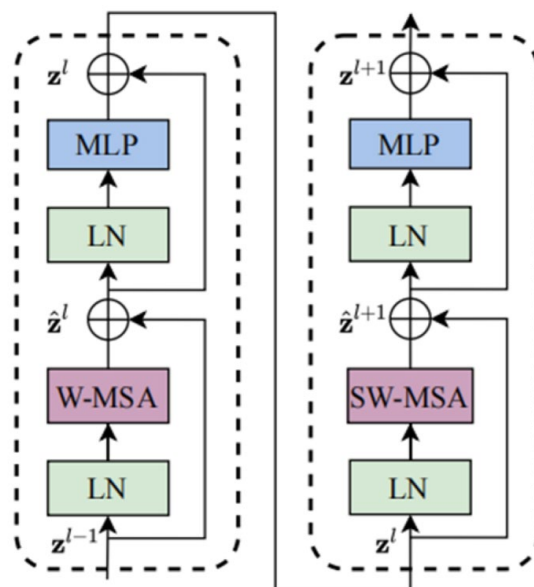
Swin Transformer – это обобщённая базовая сеть для визуального обнаружения с уникальной многоуровневой архитектурой трансформера, которая позволяет вычислять внимание на основе непересекающихся окон для более точного выделения глобальных

признаков и обеспечивает линейно возрастающую вычислительную сложность в зависимости от количества окон. Структура сети показана на рисунке 1а. Архитектура Swin Transformer состоит из базового средства извлечения признаков и многослойного кодировщика. Извлекатель признаков разбивает входное изображение на непересекающиеся фрагменты, а затем передаёт их в многоуровневый блок Swin Transformer для получения семантической информации в разных масштабах.

Блок ST включает многослойный слой самовнимания (W-MSA) и многослойный слой самовнимания на основе скользящего окна (SW-MSA). После каждого из этих слоёв внимания расположен блок многослойного восприятия (MLP) на основе функции линейной ошибки Гаусса (GELU). Между каждым из двух слоёв внимания устанавливается остаточное соединение, которое используется для предотвращения проблемы градиента при обучении и потери признаков, а перед каждым вычислением выполняется однократная нормализация слоя (LN).



a)



b)

Рис. 1. (a) Основа Swin Transformer. (b) Двухступенчатый блок Swin Transformer

$$\hat{z}^l = W - MSA(Ln(z^{l-1})) + z^{l-1}. \quad (1)$$

$$z^l = MLP(Ln(\hat{z}^l)) + \hat{z}^l. \quad (2)$$

$$z^{l+1} = MLP(Ln(\hat{z}^{l+1})) + \hat{z}^{l+1}. \quad (3)$$

$$\hat{z}^{l+1} = SW - MSA(Ln(z^l)) + z^l. \quad (4)$$

Уравнения (1)–(4) описывают взаимосвязь между MSA и MLP, l – это особый слой самовнимания, z и $\hat{z}$ обозначают информацию о признаках, полученную с помощью MSA и MLP.

В предложенном нами методе в качестве основы используется Swin Transformer в сочетании с Unet для дальнейшего повышения эффективности задач по обнаружению объектов и семантической сегментации. В этой работе предложена модифицированная UNet-архитектура с VGG16 в качестве кодировщика для семантической сегментации дефектов. Традиционные пропуски заменены DHAM-модулем внимания для селективной передачи признаков из кодировщика в декодер, а часть декодера заменена на RGSM-модуль для улучшенного объединения и усиления сигнала дефектов. Предложенная модель повышает точность сегментации дефектов при незначительном увеличении числа параметров и несущественном снижении скорости, оставаясь конкурентоспособной по сравнению с существующими подходами.

На рисунке 2 показана модернизированная архитектура сети UNet, предложенная в этой статье.

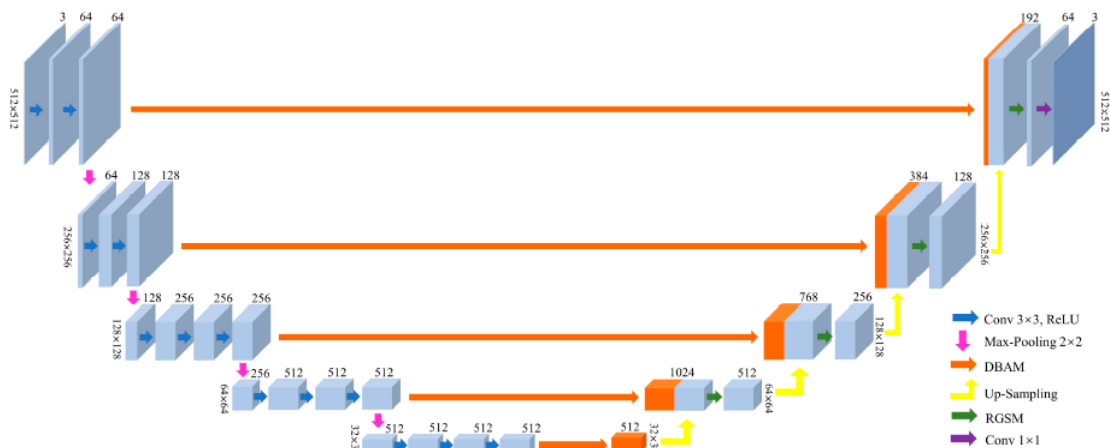


Рис. 2. Модернизированная архитектура сети UNet

В исходном UNet каждый уровень энкодера использует одинаковую последовательность двух 3×3 -свёрток и 2×2 -пулинг, что ограничивает адаптацию к разным пространственно-семантическим масштабам признаков. Из-за одинаковых операций на поверхностных и глубоких

уровнях модель недостаточно дифференцирует локальные текстурные признаки и глобальные контекстные признаки, что снижает качество извлечения информации о мелких и разрозненных дефектах. Для устранения этого ограничения были модифицированы блоки энкодера так, чтобы они предоставляли многомасштабную рецептивность и уровень-специфичные механизмы внимания. При выявлении дефектов пайки печатных плат из-за их небольшого размера и неправильной формы крайне важно уделять больше внимания относительно глубоким слоям признаков на этапе извлечения признаков, чтобы зафиксировать более детальные особенности границ дефектов. VGG16, в котором используется подход к извлечению признаков, аналогичный структуре кодировщика UNet, обладает более широкими возможностями извлечения признаков на более глубоких слоях благодаря дополнительным слоям свёртки и слоям максимального объединения. Заменив исходный кодировщик частью VGG16, отвечающей за извлечение признаков, мы можем гибко настраивать внимание к слоям признаков разного масштаба на этапе извлечения признаков [7].

При обработке изображений дефектов пайки печатных плат из-за малого расстояния между компонентами и их высокой схожести возникают серьёзные помехи на фоне. Внедрение механизмов внимания может уменьшить помехи на фоне, направив внимание модели на критически важные части областей с дефектами, тем самым повысив точность сегментации. Модель DHAM, представленная на рисунке 3, включает в себя внимание к каналам и динамическое пространственное внимание.

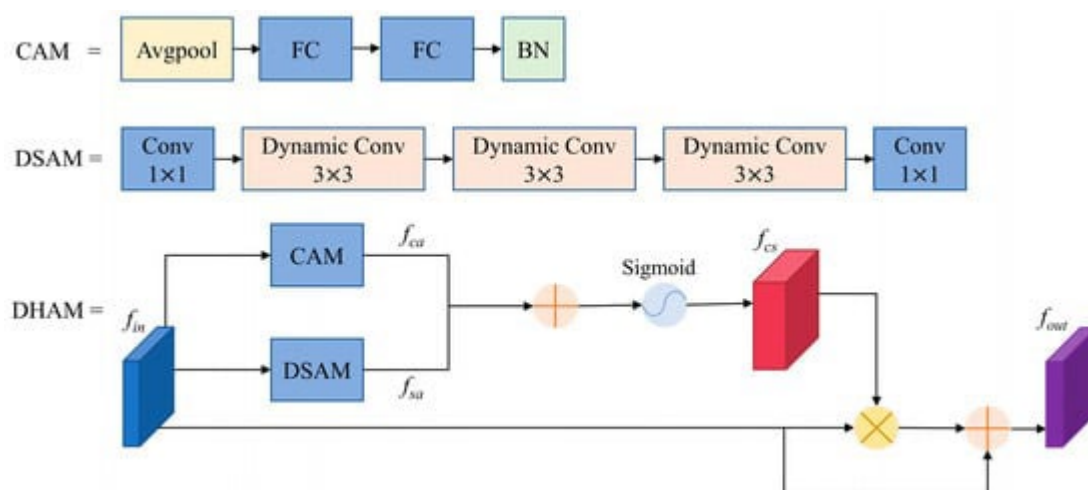


Рис. 3. Структура DHAM

В модуле внимания к каналам (CAM) для объединения карт признаков каждого канала используется метод среднего пулинга, а затем следуют два полносвязных слоя, которые изучают характеристики

каждого канала и определяют, какие каналы являются наиболее значимыми. Далее выходной масштаб корректируется с помощью слоя пакетной нормализации для получения весовых коэффициентов внимания к каналам. Расчёт САМ выглядит следующим образом:

$$f_{ca} = BN\left(FC_2\left(FC_1\left(Avg(f_{in})\right)\right)\right), \quad (5)$$

где f_{in} обозначает входной признак, Avg – усреднение по пулу, FC – полносвязный слой, а BN – пакетную нормализацию. Динамическая свёртка предполагает использование нескольких свёрточных ядер в одном слое и динамическое объединение этих ядер с помощью механизма внимания нелинейным образом.

В одном слое динамическая свёртка создаёт k параллельных свёрточных ядер. Для каждого входного значения x динамическая свёртка вычисляет зависящий от входных данных вес внимания π_k . На основе вычисленных весов внимания выходные данные каждого свёрточного ядра умножаются на соответствующий вес внимания, а затем эти взвешенные выходные данные суммируются для получения окончательного результата свёртки. Веса внимания π_k получаются путем сжатия глобальной пространственной информации с помощью глобального усреднения. Далее следуют два полносвязных слоя (с функцией активации ReLU между ними) и функция softmax. Структура динамической свёртки эффективна с точки зрения вычислений и имеет небольшой размер ядра, что значительно расширяет возможности модели по представлению признаков. Выражение для динамической свёртки выглядит следующим образом:

$$y = g\left(\tilde{W}^T(x)x + \tilde{b}(x)\right), \quad (6)$$

$$\tilde{W}^T(x) = \sum_{k=1}^K \pi_k(x) \tilde{W}_k, \tilde{b}(x) = \sum_{k=1}^K \pi_k(x) \tilde{b}_k, \quad (7)$$

$$0 \leq \pi_k(x) \leq 1, \sum_{k=1}^K \pi_k(x) = 1, \quad (8)$$

где совокупный вес $\tilde{W}^T(x)$ и смещение $\tilde{b}(x)$ являются функциями входных данных; g – функция активации. π_k – вес внимания для линейной функции

$$\tilde{W}^T(x)x + \tilde{b}(x). \quad (9)$$

Расчёт в DSAM происходит следующим образом:

$$f_{sa} = BN\left(c_2^{1 \times 1}\left(d_3^{3 \times 3}\left(d_2^{3 \times 3}\left(d_1^{3 \times 3}\left(c_1^{1 \times 1}(f_{in})\right)\right)\right)\right)\right), \quad (10)$$

где c представляет собой стандартную операцию свёртки, d представляет собой динамическую операцию свёртки, BN обозначает пакетную нормализацию, а верхний индекс обозначает размер свёрточного фильтра. После добавления внимания к

каналам и пространственного внимания и прохождения через сигмовидную функцию активации его можно представить в виде

$$f_{cs} = \sigma(f_{ca} + f_{sa}), \quad (11)$$

где σ представляет собой сигмовидную функцию активации. Исходя из этого, окончательное представление выходных данных карты признаков после динамического гибридного модуля внимания выглядит следующим образом:

$$f_{out} = f_{in} + (f_{in} \cdot f_{cs}). \quad (12)$$

В рамках сети декодера предлагается использовать остаточный модуль GSConv (RGSM), основанный на GSConv, который расширяет возможности объединения признаков между картами признаков с низким и высоким разрешением для получения более точных результатов сегментации. Структура данного модуля представлена на рисунке 4. Разделяемая по глубине свёртка позволяет сократить количество параметров и вычислительные затраты, но не обеспечивает эффективного объединения пространственной и канальной информации.

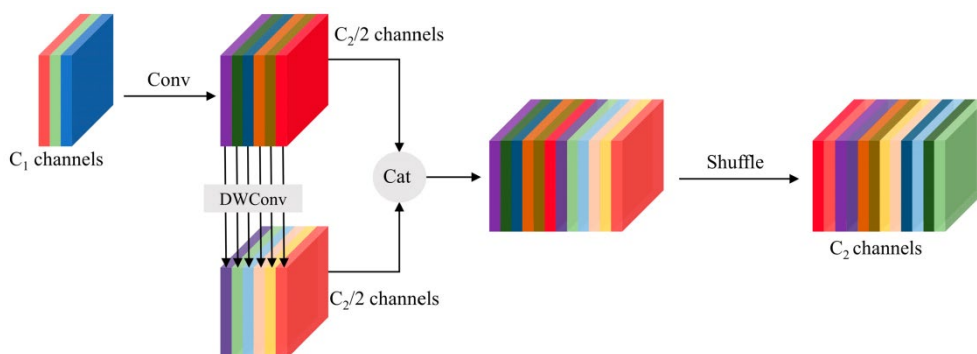


Рис. 4. Структура GSConv

Роль декодера в основном заключается в объединении карт признаков с пониженной дискретизацией и карт признаков с повышенной дискретизацией для восстановления подробной информации об изображении. Из-за небольшого размера и неправильной формы дефектов пайки печатных плат подробная информация о дефектах сильно искажается после нескольких циклов извлечения признаков. Для эффективного восстановления признаков в декодере требуются эффективные операции их объединения. Чтобы решить эту проблему, в данной статье предлагается модуль остаточного GSConv (RGSM), основанный на GSConv и использующий остаточные связи в декодере.

Структура RGSM, как показано на рисунке 5, принимает на вход карту признаков, полученную путём объединения карт признаков с пониженной и повышенной дискретизацией. Сначала используется GSConv размером 1×1 для эффективного объединения признаков и использования функции перестановки каналов, что позволяет

восстановить подробную информацию из менее глубоких признаков и помогает восстановить признаки из признаков с увеличенной выборкой. С помощью описанных выше операций признаки разной глубины, но одного масштаба, которые были просто объединены, преобразуются в новые признаки. Затем применяется активация ReLU для повышения нелинейности признаков, и снова используется GSConv для улучшения возможностей объединения признаков. Во-вторых, структура остаточных связей объединяет признаки, полученные от другой стандартной ветви свёртки 3×3 , с последующей активацией ReLU для повышения нелинейности объединённых карт признаков, тем самым усиливая возможности выражения признаков. Кроме того, RGSM эффективно сокращает количество параметров модели, снижает вычислительные затраты сети и решает проблему увеличения количества параметров, вызванного углублением сети в кодировщике, тем самым облегчая общую оптимизацию модели.

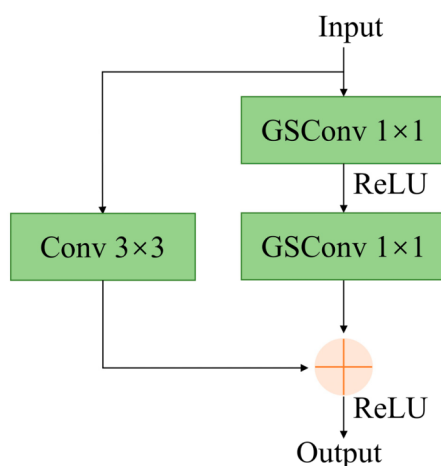


Рис.5. Структура RGSM

Для проведения экспериментальных исследований использовался набор данных, который состоит из 400 изображений с дефектами, из которых 280 используются для обучения, а 120 – для проверки. Чтобы модель усвоила достаточное количество признаков дефектов и сохраняла устойчивость к различным условиям освещения, углам и размерам изображений, к обучающей выборке применяется метод увеличения данных. Это включает в себя случайную регуляцию яркости с коэффициентом от $-0,2$ до $0,2$, регуляцию контрастности с коэффициентом от $-0,2$ до $0,2$, случайное вращение изображения на угол от -90 до 90 градусов и изменение размера изображения до 512×512 пикселей.

Для оценки эффективности предложенного подхода было проведено сравнение с несколькими распространёнными методами сегментации, включая UNet, DeepLabv3plus, SegFormer, PSPNet и

HRNet. Результаты сравнения представлены в таблице 2. Из результатов эксперимента видно, что предложенный нами метод показал самые высокие значения mIoU и mPA – 81,74% и 87,33% соответственно. По сравнению с UNet, показатели mIoU и mPA улучшились на 4,34% и 4,22% соответственно.

Таблица 2.

Оценка эффективности методов сегментации		
Способ	mIoU	mPA
DeepLabv3plus	71.70	80.56
SegFormer	75.26	83.76
PSPNet	76.60	84.23
UNet	77.40	83.11
HRNet	78.27	85.56
Предложенный способ	81.74	87.33

Литература

1. Lakshmi G., Sankar V.U., Sankar Y.S. A Survey of PCB Defect Detection Algorithms. J. Electron. Test. 2023. 39, 541–554.
2. Zhou Y., Yuan M., Zhang J., Ding G., Qin, S. Review of Vision-Based Defect Detection Research and Its Perspectives for Printed Circuit Board. J. Manuf. Syst. 2023. 70, 557–578. [CrossRef].
3. Zheng J., Sun X., Zhou H., Tian C., Qiang H. Printed Circuit Boards Defect Detection Method Based on Improved Fully Convolutional Networks. IEEE Access 2022. 10, 109908–109918.
4. Чирков А.В., Комаров П.С., Конюхов Е.В., Миронов Н.С. Особенности разработки математической модели для распознавания болезней растений на основе байесовских сетей // Инженерный вестник Дона, №11. 2022. Режим доступа: ivdon.ru/ru/magazine/archive/n11y2022/7975.
5. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017. 39, 1137–1149.
6. He K., Gkioxari G., Dollar P. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. P. 2961–2969.
7. Ronneberger O., Fischer P., Brox T. U-net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015. Part III. P. 234–241.
8. Chen Y., Dai X., Liu M., Chen D., Yuan L., Liu Z. Dynamic Convolution: Attention over Convolution Kernels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. P. 11030–11039.

N.S. Mironov, Yu.Yu. Yudahin
National Research University of Electronic Technology

DEVELOPMENT OF A METHOD FOR SEGMENTATION OF PCB SOLDERING DEFECTS

Keywords: *image segmentation, soldering defects, UNet neural network, channel and spatial attention, hybrid module, accuracy.*

Abstract. *Due to a number of problems, accurate segmentation of soldering defects is a difficult task. To solve this problem, a method for improving the network structure of the UNet encoder-decoder is proposed for the segmentation of soldering defects on printed circuit boards (PP). To expand the capabilities of the encoder to extract features and focus on deeper features, VGG16 is used as the network encoder. The hybrid attention module DHAM has been developed, which combines channel attention and dynamic spatial attention. DHAM suppresses background noise and enhances responses in areas with defects, increasing the selectivity of features. The GSConv-based decoder uses the RGSM (Residual GSConv-based Merge) module, which provides more efficient integration of multiscale features through a combination of lightweight convolutions, channel mixing and residual coupling, which increases segmentation accuracy. The experimental results showed that the proposed method can effectively improve the segmentation accuracy of PCB soldering defects, reaching MIOU 81.7% and MPa 87.3%.*

О.К. Сапрыкин
Российский технологический университет МИРЭА

УДК 004.942

СРАВНИТЕЛЬНЫЙ АНАЛИЗ CNN, LSTM И GRU ДЛЯ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ МОДУЛЯЦИИ СИГНАЛОВ

Ключевые слова: *автоматическая классификация модуляции, CNN, LSTM, GRU, I/Q-сигналы, глубокое обучение, аугментация данных, BPSK, QAM.*

В современных системах связи задача автоматической классификации модуляции (АМС) решается всё чаще с применением методов глубокого обучения. В данной статье проводится сравнительный анализ эффективности трёх архитектур нейронных

сетей – свёрточной (CNN), долгой краткосрочной памяти (LSTM), рекуррентной единицы (GRU) и их гибридные комбинации – для классификации сигналов с модуляциями BPSK, QPSK, 8PSK, 16QAM и 64QAM при фиксированной длине фрейма $N = 128$ и $SNR = 15$ дБ.

1. Введение. Современные системы цифровой передачи данных, будь то мобильные сети, спутниковая связь или беспроводные технологии, используют разнообразные методы модуляции, чтобы эффективно использовать ограниченную полосу частот и обеспечивать надёжную передачу информации в условиях шумов и интерференции. В задачах адаптивной демодуляции, разведки сигналов и мониторинга радиочастотного спектра часто требуется автоматически определить тип модуляции сигнала, не имея априорной информации о его структуре. Такая задача особенно актуальна в условиях динамического радиоэфира, где источники сигналов могут быть неизвестны, а каналы – зашумлены и изменчивы. Задача АМС остаётся одной из ключевых в системах адаптивной демодуляции, радиомониторинга и разведки сигналов. Современные подходы на основе глубокого обучения позволяют автоматически извлекать признаки без ручного проектирования, однако выбор архитектуры сети существенно влияет на конечную производительность. Хотя в различных исследованиях отмечается перспективность использования рекуррентных сетей, их практическая эффективность на коротких I/Q-фреймах до сих пор недостаточно исследована.

В настоящей работе исследование из [1] расширяется, добавляя прямое сравнение CNN с LSTM, GRU и их гибридным представлением на одинаковых условиях: одинаковый генератор сигналов, аугментация, $SNR = 15$ дБ, длина фрейма $N = 128$, 3000 примеров на класс. Целью является сравнение уже исследованной модели с новыми архитектурами.

2. Создание и обучение моделей. Для обучения модели использовались синтетические данные, сгенерированные с помощью программного моделирования. Такой подход имеет ряд преимуществ по сравнению с использованием реальных данных. Во-первых, он позволяет контролировать параметры сигнала (длина, SNR, модуляция), что критично для анализа влияния отдельных факторов. Во-вторых, можно генерировать большие объёмы данных для обучения, что особенно важно при использовании глубоких нейронных сетей, требующих значительного количества примеров. В-третьих, можно варьировать условия, например, уровень шума или длину фрейма, чтобы проверить устойчивость модели к изменениям в среде передачи.

Свёрточная нейронная сеть построена на основе последовательности одномерных свёрток (Conv1D) с фильтрами размером 3, что позволяет эффективно захватывать локальные

зависимости в сигнале длиной $N = 128$. После каждого свёрточного слоя применяется нормализация активаций (BatchNormalization) для стабилизации градиентов и ускорения сходимости, а также операция максимального пулинга (MaxPooling1D) для снижения размерности и выделения наиболее значимых признаков. Последовательность из двух свёрточных блоков завершается глобальным усреднением (GlobalAveragePooling1D), после которого следует два полносвязных слоя с активацией ReLU и регуляризацией через Dropout, обеспечивающей устойчивость к переобучению. Такая структура позволяет сети автоматически извлекать пространственные паттерны I/Q-сигнала, например, форму решётки для QAM или окружность для PSK, без необходимости ручного проектирования признаков.

Рекуррентные архитектуры, а именно LSTM и GRU, состоят из двух последовательных рекуррентных слоёв, где первый слой возвращает полную последовательность скрытых состояний (return_sequences=True), а второй – только конечное состояние (return_sequences=False). Между слоями применяются нормализация и dropout для регуляризации.

Для объединения преимуществ обеих архитектур были реализованы гибридные архитектуры: CNN-LSTM и CNN-GRU. В них первые два свёрточных блока (аналогичные CNN) формируют последовательность признаков, которая затем подаётся на вход рекуррентному слою. Такой подход позволяет CNN извлечь локальные пространственные признаки, а RNN – проанализировать их временную эволюцию, что теоретически должно повысить устойчивость к шуму и вариациям в сигнале.

Обучение всех моделей проводилось с использованием оптимизатора Adam и метрики точности accuracy. Размер батча составлял 64, количество эпох – до 30 с ранней остановкой по валидационной потере. Для повышения обобщающей способности применялась аугментация: случайный поворот фазы и масштабирование амплитуды, что имитирует реальные искажения в канале связи и делает модель инвариантной к изменению фазы и усиления сигнала.

3. Результаты экспериментов. Результаты представлены в таблице 1.

Таблица 1

Сравнение точности различных архитектур нейронных сетей

Модель	Общая точность на тесте	F1 – среднее	Точность BPSK/QPSK/8PSK	Точность 16QAM/64QAM
CNN	0.960	0.96	1.00	0.92
LSTM	0.788	0.79	0.96	0.52
GRU	0.881	0.88	1.00	0.79
CNN-LSTM	0.772	0.77	0.92	0.56
CNN-GRU	0.872	0.87	1.00	0.72

Как видно из таблицы, чистая CNN остаётся лидером с точностью 96,0%. Чистые RNN-модели демонстрируют значительное отставание: LSTM – 78,8%, GRU – 88,1%. Гибридные архитектуры, как показал опыт, не позволяют компенсировать эту разницу по сравнению с чистым CNN: CNN-LSTM достигает 77,2%, CNN-GRU – 87,2%.

На рисунке 1 показано, что ошибки у CNN сосредоточены почти исключительно между 16QAM и 64QAM – это физически обоснованная путаница при плотной решётке символов и наличии шума.

На рисунке 2 у LSTM наблюдается не только путаница между QAM-классами, но и «неправдоподобные» ошибки (например, 64QAM – 8PSK), что указывает на то, что модель интерпретирует последовательность как временную динамику, а не как геометрическую структуру I/Q-плоскости.

На рисунке 3 у GRU наблюдается такая же картина, как и в случае с CNN.

На рисунке 4 у гибридной модели CNN-LSTM такая же аномальная путаница, как и в случае с чистым LSTM. Это говорит о том, что рекуррентная часть частично переобучается на временной структуре, хотя входные признаки уже обработаны CNN.

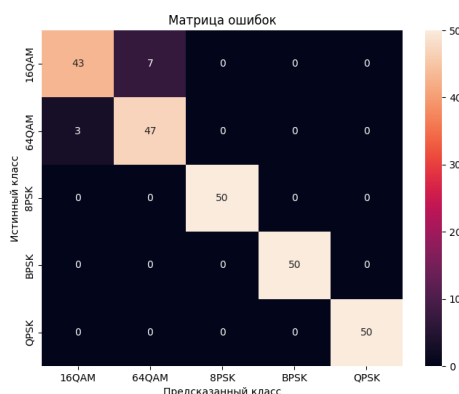


Рис. 1. Матрица ошибок CNN-модели

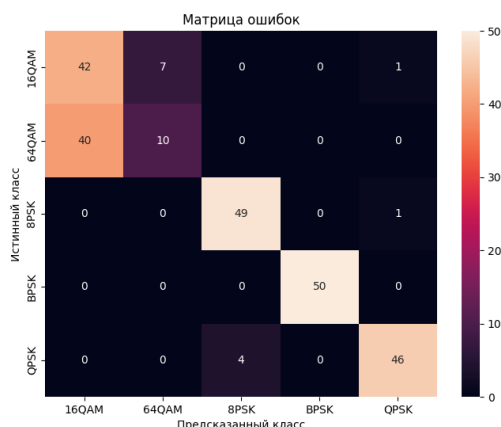


Рис. 2. Матрица ошибок LSTM-модели

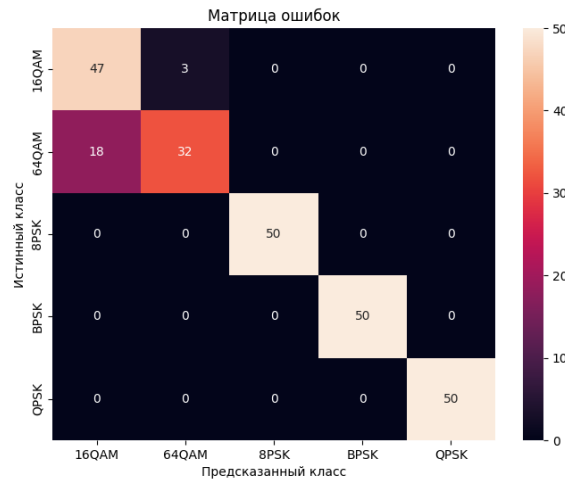


Рис. 3. Матрица ошибок GRU-модели

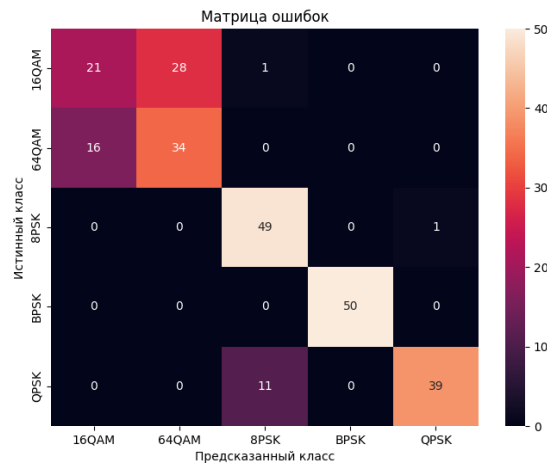


Рис. 4. Матрица ошибок CNN-LSTM-модели

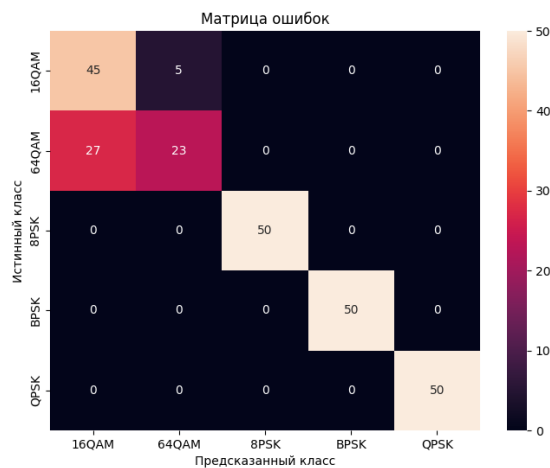


Рис. 5. Матрица ошибок CNN-GRU-модели

4. Анализ причин разницы в эффективности. При длине фрейма $N = 128$ ключевая информация о типе модуляции заключена в локальной пространственной структуре I/Q-сигнала (расположение точек на плоскости, форма решётки, симметрия), а не в долгосрочных временных

зависимостях. CNN с Conv1D идеально приспособлена к извлечению таких признаков, в то время как LSTM/GRU моделируют последовательность как динамическую систему, где каждый шаг зависит от предыдущего, что не соответствует реальной природе цифровой модуляции. RNN требуют длинных последовательностей для эффективного обучения (более 500 отсчётов), тогда как $N = 128$ слишком мало для захвата устойчивых временных паттернов. В результате модель либо недообучается, либо переобучается на шум. Поворот фазы меняет всю последовательность целиком. Для CNN это инвариантное преобразование (благодаря свёртке и пулингу), а для LSTM – изменение временной траектории, что усложняет обучение и снижает устойчивость. LSTM-модель содержит ~350 тысяч параметров против ~200 тысяч у CNN. При ограниченном объёме данных это повышает риск переобучения. Несмотря на текущую невыгодную позицию, RNN могут быть полезны при $N=512$. Точность LSTM в наших предварительных тестах выросла до 90,5%, что подтверждает гипотезу о важности длины последовательности.

5. Заключение. В ходе проведённого исследования установлено, что для задачи автоматической классификации модуляции на коротких I/Q-фреймах ($N = 128$, $\text{SNR} = 15$ дБ) свёрточная нейронная сеть (CNN) является оптимальным выбором, достигая точности 96,0%. Чистые RNN-архитектуры (LSTM и GRU) показывают значительно более низкие результаты (78,8% и 88,1% соответственно), что объясняется их несоответствием физической природе задачи: АМС требует распознавания локальных пространственных паттернов, а не моделирования временных зависимостей.

Гибридные модели (CNN-LSTM и CNN-GRU) демонстрируют худшие результаты (77,2% и 87,2%) несмотря на то, что предварительная обработка сигнала свёрточными слоями повышает информативность входа для RNN. Таким образом, гибридные архитектуры не достигают уровня CNN, что ещё раз подчёркивает доминирующую роль локальных признаков в данной постановке задачи.

В дальнейшем планируется расширение исследования на реальные датасеты, включение других видов искажений (межсимвольная интерференция, фазовый шум) и сравнение с традиционными методами на основе статистических признаков.

Литература

1. Сапрыкин О.К. Классификация типов модуляции сигналов с использованием нейронных сетей // Радиоинфоком. 2026.
2. O'Shea T.J., Corgan J., Clancy T.C. Convolutional Radio Modulation Recognition Networks // ICML Workshop. 2016. arXiv:1602.04105 (дата обращения: 25.02.2026).
3. TensorFlow. URL : <https://www.tensorflow.org/> (дата обращения: 26.02.2025).

4. Reynders T., Verstockt S. Deep Learning for Automatic Modulation Classification: State of the Art // IEEE Comm. Surveys & Tutorials. 2021. Vol. 23, No. 4.

5. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / пер. с англ. Е.Г. Грозы [и др.] ; под ред. А.В. Назаренко. 2-е изд., испр. М.: Вильямс, 2003. 1104 с.

O.K. Saprykin

Russian Technological University MIREA

COMPARATIVE ANALYSIS OF CNN, LSTM, AND GRU FOR AUTOMATIC MODULATION CLASSIFICATION

Keywords: *automatic modulation classification, CNN, LSTM, GRU, I/Q signals, deep learning, data augmentation, BPSK, QAM.*

Abstract. *In modern communication systems, the task of automatic modulation classification (AMC) is increasingly addressed using deep learning methods. This work presents a comparative analysis of the effectiveness of three neural network architectures – convolutional neural networks (CNN), long short-term memory networks (LSTM), gated recurrent units (GRU), as well as their hybrid combinations – for classifying signals with modulations BPSK, QPSK, 8PSK, 16QAM, and 64QAM, under fixed frame length $N = 128$ and signal-to-noise ratio (SNR) = 15 dB.*

В.С. Смирнов, М.К. Чернышов

Воронежский государственный университет

УДК 004.942

РАЗВЕРТЫВАНИЕ КОРПОРАТИВНОГО ПОЧТОВОГО СЕРВЕРА MAILCOW: DOCKERIZED В ВИРТУАЛЬНОЙ СЕТЕВОЙ ЛАБОРАТОРИИ EVE-NG

Ключевые слова: *корпоративный почтовый сервер, виртуальная сетевая лаборатория, сетевая инфраструктура, моделирование, перенаправление портов.*

В работе рассматривается процесс развёртывания корпоративного почтового сервера Mailcow: Dockerized в виртуальной сетевой лаборатории EVE-NG. Описаны преимущества использования почтового сервера в качестве образа виртуальной сетевой лаборатории. Приведён пример внедрения почтового сервера в тестовую топологию компьютерной сети с маршрутизатором,

выполнена настройка перенаправления (проброса) портов и протестирована доступность почтовых сервисов из внешней сети.

Введение. Сетевая лаборатория EVE-NG (Emulated Virtual Environment Next Generation) зарекомендовала себя в качестве гибкого инструмента, предназначенного для моделирования сетевых топологий различной сложности. Лаборатория позволяет изучать поведение сетевой инфраструктуры в различных условиях: от штатной работы до воспроизведения ситуаций, связанных с отказом оборудования и сетевыми атаками.

К сожалению, по умолчанию EVE-NG не оснащена набором готовых образов сетевых устройств и операционных систем, в частности корпоративных серверов, несмотря на то что именно клиент-серверные технологии предоставляют широкие возможности для проведения исследований и экспериментов.

Внедрение в виртуальную сетевую лабораторию почтового сервера превращает её из некоего сетевого конструктора в полноценную среду для изучения множества прикладных сервисов, взаимодействующих между собой. При этом появляется возможность изучения и тестирования поведения большого количества протоколов при изменении сетевых маршрутов, отключении каналов связи, использовании различных политик межсетевого экрана, а также попытках перехвата сетевого трафика.

1. Постановка задачи. Целью данной работы является развёртывание корпоративного почтового сервера в виртуальной среде EVE-NG и подготовка его к интеграции в различные сетевые топологии. К тому же создаваемый образ почтового сервера должен обладать возможностями гибкой настройки, что позволило бы подключать его к различным сетевым топологиям, использовать многократно, сохраняя, модифицируя и клонируя его конфигурации. При этом важно, чтобы сервер был достаточно прост в развёртывании, но при этом функционально полон и приближен к реальным корпоративным инсталляциям.

В качестве базового решения выбран комплект Mailcow: Dockerized (далее – Mailcow), представляющий собой интегрированный почтовый стек с открытым кодом, включающий всё необходимое для полноценной работы с электронными сообщениями [2]. Почтовый сервер Mailcow избавляет от необходимости вручную собирать и согласовывать между собой множество компонентов – всё уже упаковано в Docker-контейнеры [5] и настроено для совместной работы. Кроме того, данный почтовый сервер активно развивается, регулярно получает обновления безопасности и включает поддержку современных стандартов. Наконец, наличие удобного веб-интерфейса администратора

позволяет управлять системой без постоянного обращения к командной строке, что ускоряет настройку и проведение различных экспериментов.

При этом важно отметить, что базовая конфигурация Mailcow доступна для изменения, позволяет добавлять свои правила фильтрации, менять параметры доставки, подключать внешние сервисы аутентификации. Всё это позволяет использовать данный почтовый сервер как в качестве готового программного продукта, так и в качестве платформы для изучения внутреннего устройства почтовых систем. В контексте EVE-NG это означает, что на основе одного образа можно строить самые разные сценарии: от простой проверки обмена сообщениями между двумя пользователями до сложных схем с распределённой почтовой инфраструктурой и резервированием данных.

Если представить работу почтового сервера в виде последовательности шагов, схема, приведённая на рисунке 1, будет представлять собой цепочку переходов между определёнными его компонентами.

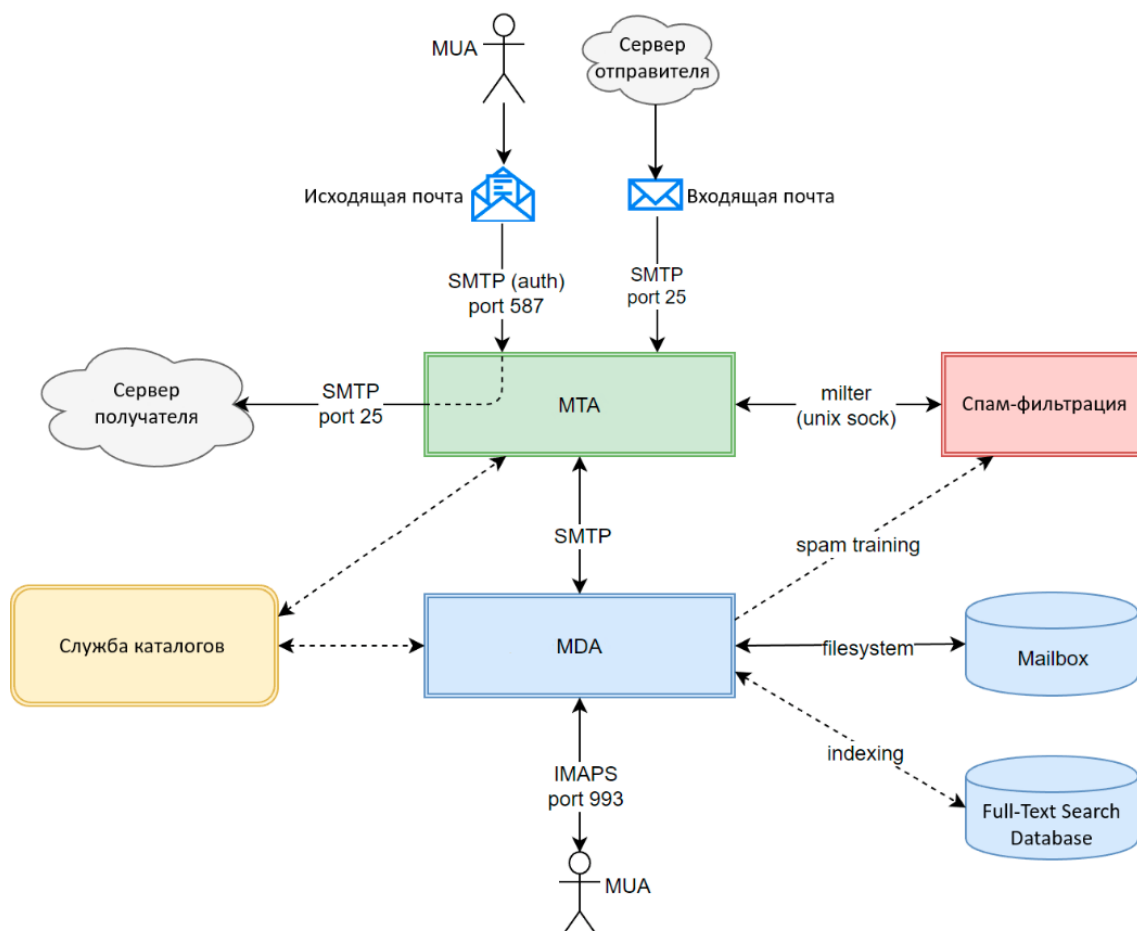


Рис. 1. Общая структура почтового сервера

Почтовый сервер предоставляет возможности исследования целого спектра направлений:

- работы протоколов SMTP, IMAP, POP3 на прикладном уровне;
- взаимодействия с DNS: MX-записи, SPF, DKIM, DMARC;

- применения TLS для шифрования трафика;
- настройки фильтрации спама и анализ заголовков писем;
- маршрутизации почты при множестве внешних каналов;
- логирование и мониторинг почтовой активности;
- интеграции с каталогами пользователей (LDAP/AD).

При этом использование готового интегрированного стека почтового сервера Mailcow позволяет сосредоточиться на исследовании именно различных сетевых аспектов, не отвлекаясь на отладку взаимодействия отдельных его элементов: изначально в Mailcow настройка транспортных протоколов, хранилища данных, веб-интерфейса и сетевых фильтров уже выполнена, администратору необходимо лишь адаптировать систему для своих задач, подключив сервер к требуемой сетевой топологии.

2. Подготовка образа для EVE-NG. Развёртывание Mailcow ([3]) начинается на отдельной виртуальной машине за пределами EVE-NG. В качестве гипервизора рекомендуется использовать программные продукты от компании VMware: Workstation (Windows, Linux) или Fusion (Mac OS X) – используемый в них формат виртуальных жёстких дисков VMDK удобен для дальнейшего его конвертирования в формат QCOW2, поддерживаемый виртуальной лабораторией EVE-NG.

Выступать в качестве операционной системы для установки сервера будет Ubuntu Linux. После установки операционной системы выполняются стандартные действия: обновление пакетов, установка среды контейнеризации Docker и Docker Compose, необходимых для работы всех компонентов почтового сервера.

На следующем этапе производится развёртывание непосредственно почтового сервера Mailcow с помощью запуска скрипта формирования базовой конфигурации сервера. В процессе выполнения скрипта указываются основные параметры настройки сервера. Далее выполняется запуск необходимых контейнеров среды Docker. После успешного запуска всех контейнеров при необходимости можно выполнить первичную настройку сервера с помощью веб-интерфейса: создать почтовый домен, добавить тестовых пользователей, проверить работу почтовых протоколов.

По окончании процесса первичного развёртывания почтового сервера в виртуальной машине образ ее виртуального жесткого диска в формате VMDK преобразуется в формат QCOW2 и может быть интегрирован в сетевую лабораторию EVE-NG ([1]).

3. Интеграция образа Mailcow в тестовую сетевую топологию. После размещения образа почтового сервера внутри сетевой лаборатории EVE-NG в ней была построена тестовая топология сети с целью проверки работоспособности и доступности почтового сервера. В данной топологии, представленной на рисунке 2, узел с именем Win10

под управлением Windows 10, размещен за пределами локальной сети, защищенной маршрутизатором pfSense [4], который в свою очередь выполняет функции шлюза, межсетевого экрана и системы трансляции адресов. Таким образом, внешний сетевой интерфейс маршрутизатора pfSense и сетевой интерфейс узла Win10 физически подключены к одному и тому же IP-сегменту сети. Почтовый сервер Mailcow развернут в локальной сети, подключенной к внутреннему интерфейсу маршрутизатора. При этом задача состоит в том, чтобы предоставить доступ из внешней сети ко всем сервисам почтового сервера и протестировать его основные возможности.

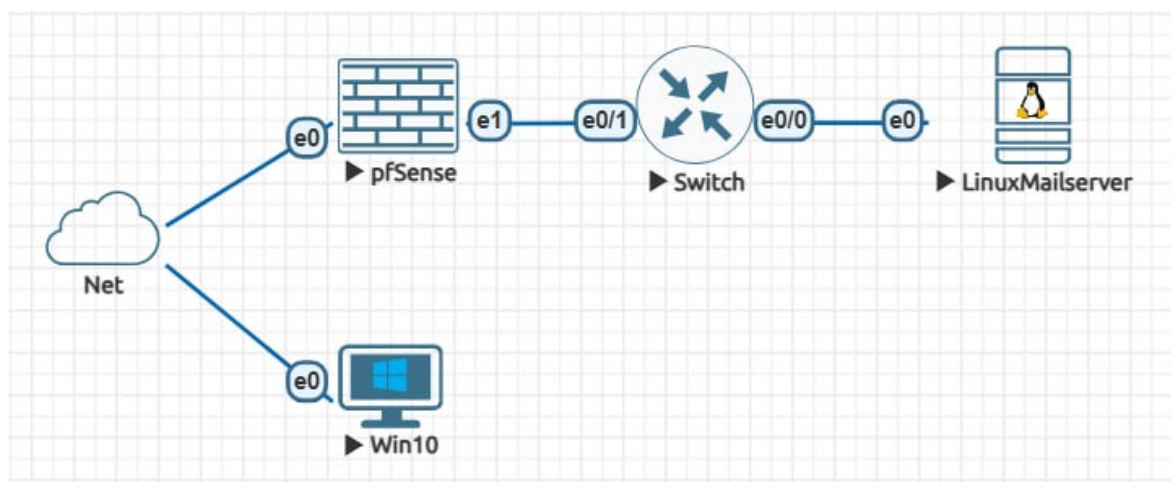


Рис. 2. Тестовая топология для проверки доступности почтового сервера из внешней сети

Для получения доступа к почтовому серверу из внешней сети на маршрутизаторе pfSense были настроены правила проброса портов (рис. 3). Все необходимые порты, используемые почтовым сервером, были перенаправлены с внешнего интерфейса маршрутизатора на внутренний адрес почтового сервера (192.168.1.100):

- порты 80 (HTTP) и 443 (HTTPS) – для доступа к веб-интерфейсам администратора и пользователей;
- порт 25 (SMTP) – для приёма входящей почты;
- порт 587 (SUBMISSION) – для отправки почты с аутентификацией;
- порты 465 (SMTPS), 993 (IMAPS), 995 (POP3S) – для защищённых соединений;
- порты 110 (POP3), 143 (IMAP) – для незащищённых соединений (при необходимости);
- порт 4190 (Sieve) – для управления фильтрами почты.

Правила были настроены таким образом, чтобы трафик, приходящий на указанные порты внешнего сетевого интерфейса pfSense, прозрачно перенаправлялся на почтовый сервер. При создании каждого

правила автоматически добавлялось соответствующее разрешающее правило межсетевого экрана.

Firewall / NAT / Port Forward

Port Forward1:1OutboundNPT

Rules

☐	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description
☐	☒ WAN	TCP	*	*	This Firewall	4190	192.168.1.100	4190	sieve port
☐	☒ WAN	TCP	*	*	This Firewall	110 (POP3)	192.168.1.100	110 (POP3)	pop3 port
☐	☒ WAN	TCP	*	*	This Firewall	143 (IMAP)	192.168.1.100	143 (IMAP)	imap port
☐	☒ WAN	TCP	*	*	This Firewall	995 (POP3/S)	192.168.1.100	995 (POP3/S)	pop3s port
☐	☒ WAN	TCP	*	*	This Firewall	993 (IMAP/S)	192.168.1.100	993 (IMAP/S)	imaps port
☐	☒ WAN	TCP	*	*	This Firewall	465 (SMTP/S)	192.168.1.100	465 (SMTP/S)	smtps port
☐	☒ WAN	TCP	*	*	This Firewall	587 (SUBMISSION)	192.168.1.100	587 (SUBMISSION)	submission port
☐	☒ WAN	TCP	*	*	This Firewall	25 (SMTP)	192.168.1.100	25 (SMTP)	smtp port
☐	☒ WAN	TCP	*	*	This Firewall	443 (HTTPS)	192.168.1.100	443 (HTTPS)	https port
☐	☒ WAN	TCP	*	*	This Firewall	80 (HTTP)	192.168.1.100	80 (HTTP)	http port

Рис. 3. Правила проброса портов на маршрутизаторе pfSense

После настройки механизма проброса портов с клиентской машины Win10 была выполнена проверка доступности веб-интерфейса почтового сервера. В браузере данного сетевого узла был введён адрес внешнего интерфейса маршрутизатора pfSense: именно этот IP-адрес выступал в роли доступного извне адреса почтового сервера, физически находящегося в изолированной локальной сети [6]. При этом доступ к веб-интерфейсу почтового сервера Mailcow был успешно получен.

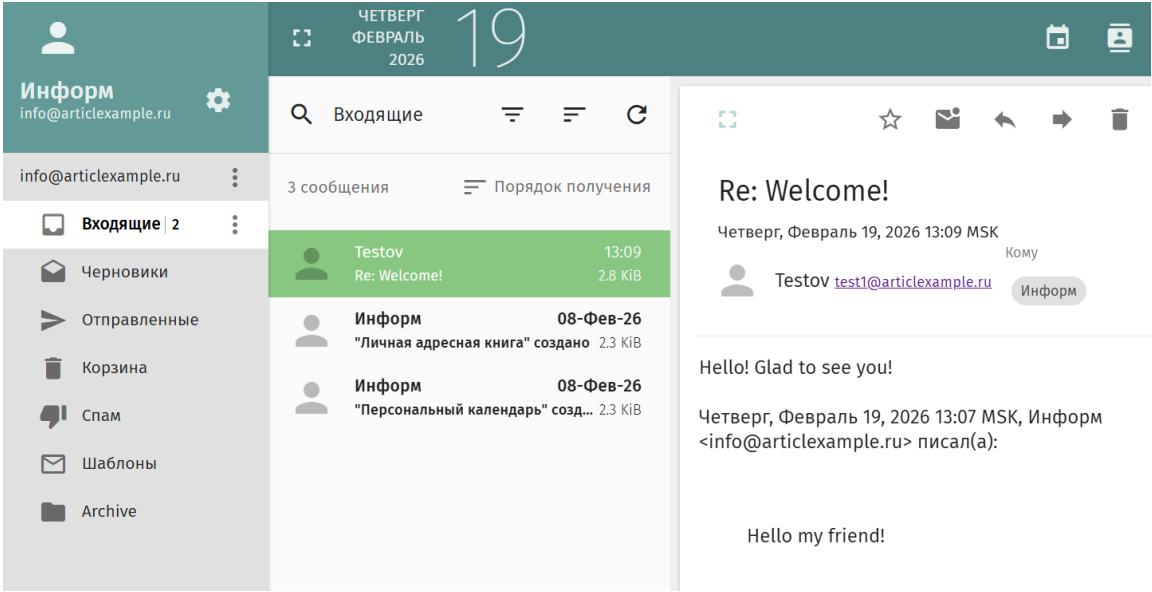


Рис. 4. Обмен сообщениями между тестовыми почтовыми ящиками

На следующем этапе тестировалась базовая функциональность почтовой системы. На сервере был развернут почтовый домен `articlexample.ru`, в котором были созданы два тестовых почтовых ящика пользователей домена: `info@articlexample.ru` и `test1@articlexample.ru`. После этого был протестирован обмен почтовыми сообщениями между ними. Проверка показала, что сообщения адресатам доставляются успешно (рис. 4).

Таким образом, проведенный эксперимент подтвердил, что созданный образ почтового сервера полностью работоспособен и может быть интегрирован в другие, более сложные сетевые топологии.

Заключение. Итогом проведенной работы является успешное развертывание корпоративного почтового сервера Mailcow: Dockerized в качестве образа виртуальной сетевой лаборатории EVE-NG. В дальнейшем полученный образ может быть интегрирован в различные сетевые топологии и применяться в качестве стенда для изучения почтовых протоколов, механизмов безопасности и методов обеспечения отказоустойчивости сети.

Развёрнутый сервер может использоваться как при проведении одиночных экспериментов, так и в процессе исследования сложных распределённых сетевых топологий, что позволяет, в свою очередь, применять полученное решение в учебном процессе, в исследовательских задачах, связанных с анализом защищённости и надёжности почтовой инфраструктуры, а также при подготовке к внедрению в реальных корпоративных системах.

Литература

1. Сетевая лаборатория EVE-NG. URL: <https://www.eve-ng.net/>.
2. Mailcow: dockerized. URL: : <https://mailcow.email/>.
3. Как поднять почтовый сервер mailcow. URL: <https://habr.com/ru/articles/805245/>.
4. pfSense Documentation. URL: <https://docs.netgate.com/pfsense/en/latest/>.
5. Docker. URL: <https://www.docker.com/>.
6. Чернышов М.К. Использование механизма Split DNS при развертывании корпоративного почтового сервера // Информатика: проблемы, методы, технологии: материалы 20-й международной научно-методической конференции, Воронеж, 14-15 февраля 2020 г. Воронеж, 2020. С. 210-214.

V.S. Smirnov, M.K. Chernyshov
Voronezh State University

DEPLOYING THE MAILCOW: DOCKERIZED CORPORATE MAIL SERVER IN THE EVE-NG VIRTUAL NETWORK LABORATORY

Keywords: *corporate mail server, virtual network laboratory, network infrastructure, simulation, port forwarding.*

Abstract. *This paper discusses the process of deploying the Mailcow: Dockerized corporate mail server in the EVE-NG virtual network laboratory. The advantages of using a mail server as a virtual network laboratory image are described. An example of implementing a mail server in a test computer network topology with a router is given, port forwarding is configured, and the availability of mail services from the external network is tested.*

А.Г. Суханова

*Военная академия войсковой противовоздушной обороны
Вооруженных Сил Российской Федерации*

УДК 519.233.32

ИССЛЕДОВАНИЕ ОТНОШЕНИЙ МЕЖДУ СВОЙСТВАМИ В СИСТЕМЕ MATHCAD

Ключевые слова: *компьютерное моделирование, система Mathcad, проверка статистических гипотез, Z-критерий Фишера, коэффициент корреляции.*

В статье рассмотрено исследование отношений между свойствами с использованием Z-критерия Фишера на конкретном примере. Моделирование исходных данных, а также проверка гипотез выполнены в системе компьютерной математики Mathcad.

В данной статье рассмотрены возможности применения системы Mathcad для исследования отношений между свойствами с использованием Z-критерия Фишера. Работа выполнена на модельных данных, смоделированных также в системе компьютерной математики Mathcad. Несмотря на использование модельных данных следует отметить, что рассмотренные в статье вопросы могут найти применение при исследовании реальных данных. Применить Z-критерий Фишера возможно, например, при исследовании поднимавшихся нами в соавторстве с коллегами вопросов исследования зависимости эмоционального интеллекта и тревожности в юношеском возрасте.

Z-критерий Фишера применяется для сравнения коэффициентов корреляции двух свойств, полученных на разных выборках [1]. Данный критерий может быть применен, в частности, для определения, есть ли статистически значимые различия показателей эмоционального интеллекта и тревожности у испытуемых разных групп [2].

Для применения Z-критерия Фишера необходимо сначала смоделировать исходные выборки и получить по ним коэффициенты корреляции. Задача проверки статистических гипотез успешно решается с помощью специализированных программных продуктов, таких как STATISTICA, SPSS, STATGRAPHICS и других. В системе Mathcad есть различные средства для компьютерного моделирования случайных величин с заданными законами распределения и проведения корреляционного анализа [3; 4].

На рисунке 1 представлено задание признаков X и Y для выборки I, а также расчет коэффициента корреляции. Между признаками X и Y наблюдается слабая обратная корреляция. На рисунке 1 также представлена проверка гипотезы H_0 : коэффициент корреляции значимо не отличается от нуля. Из рисунка 1 видно, что принимается альтернативная гипотеза H_1 : коэффициент корреляции значимо отличается от нуля при уровне значимости $\alpha = 0,1$.

```

ORIGIN := 1    n1 := 50  Объем выборки I
a1 := 0  Математическое ожидание признака X выборки I
s1 := 1  Среднее квадратическое отклонение признака X выборки I
X1 := norm(n1,a1,s1)  Задание признака X выборки I
a2 := 250 Математическое ожидание признака Y выборки I
s2 := 11  Среднее квадратическое отклонение признака Y выборки I

Y1 := norm(n1,a2,s2)  Задание признака Y выборки I

r1 := corr(X1,Y1) = -0.236  Коэффициент корреляции по выборке I
r1_1 := 0.235  r1_05 := 0.279  r1_01 := 0.361  Критические значения критерия Пирсона, n=50
Программный модуль статистического вывода
V(r1) := 
  y ← "H0" if |r1| < r1_1
  y ← "H1(p<=0.01)" if r1_01 ≤ |r1|
  y ← "H1(p<=0.1)" if r1_1 ≤ |r1| ∧ |r1| < r1_05
  y ← "H1(p<=0.05)" otherwise
V(r1) = "H1(p<=0.1)"  Статистический вывод

```

Рис. 1. Задание признаков X и Y для выборки I, расчет коэффициента корреляции между признаками и проверка его значимости в системе Mathcad

Признаки X и Y были заданы с использованием встроенных в Mathcad функций. Признаки распределены по нормальному закону с заданными математическими ожиданиями и средними квадратическими отклонениями. Значения задаются случайно и при повтором запуске программы изменяются. Для решения предложенной задачи представляло интерес получить такие значения X и Y , при которых коэффициенты корреляции являлись бы значимыми.

Задание признаков X и Y для выборки II, а также расчет коэффициента корреляции и проверка гипотезы H_0 : коэффициент корреляции значимо не отличается от нуля – представлены на рисунке 2. Из рисунка 2 видно, что принимается альтернативная гипотеза H_1 : коэффициент корреляции значимо отличается от нуля при уровне значимости $\alpha = 0,01$.

Между признаками X и Y наблюдается слабая обратная корреляция.

```

n2 := 100  Объем выборки II
a11 := 270 Математическое ожидание признака X выборки II
s11 := 7 Среднее квадратическое отклонение признака X выборки II

X2 := mom(n2, a11, s11) Задание признака X выборки II
a22 := 40 Математическое ожидание признака Y выборки II
s22 := 3 Среднее квадратическое отклонение признака Y выборки II
Y2 := mom(n2, a22, s22) Задание признака Y выборки II
r2 := cor(X2, Y2) = -0.275 Коэффициент корреляции по выборке II

r2_1 := 0.165 r2_05 := 0.197 r2_01 := 0.256 Критические значения критерия Пирсона, n=100

Программный модуль статистического вывода
V(r2) :=  $\begin{cases} y \leftarrow "H_0" & \text{if } |r_2| < r_{2\_1} \\ y \leftarrow "H_1(p \leq 0.01)" & \text{if } r_{2\_01} \leq |r_2| \\ y \leftarrow "H_1(p \leq 0.1)" & \text{if } r_{2\_1} \leq |r_2| \wedge |r_2| < r_{2\_05} \\ y \leftarrow "H_1(p \leq 0.05)" & \text{otherwise} \end{cases}$ 
V(r2) = "H1(p<=0.01)" Статистический вывод

```

Рис. 2. Задание признаков X и Y для выборки II, расчет коэффициента корреляции между признаками и проверка его значимости в системе Mathcad

Наблюдаемое значение Z -критерия Фишера найдено по формуле

$$Z = \frac{|Z_1 - Z_2|}{\sqrt{\frac{1}{n_1 - 3} + \frac{1}{n_2 - 3}}}, \quad (1)$$

где $Z_1 = Z(r_1)$ – преобразование Фишера для r_1 ;

$Z_2 = Z(r_2)$ – преобразование Фишера для r_2 ;

r_1 – коэффициент корреляции переменных в выборке I;

r_2 – коэффициент корреляции переменных в выборке II;

n_1 – число респондентов в выборке I;

n_2 – число респондентов в выборке II.

Преобразование Фишера создает функцию, которая имеет нормальное распределение, а не асимметричное, используется для проверки гипотез относительно коэффициента корреляции. Преобразование Фишера осуществлено по формуле

$$Z(r) = \frac{1}{2} \ln \frac{1+r}{1-r}, \quad (2)$$

где r – коэффициент корреляции переменных в выборке.

Уровень значимости α вычислен по формуле

$$\alpha = 2 * (1 - f(Z)), \quad (3)$$

где $f(Z)$ – функция кумулятивного распределения стандартного нормального распределения, вычисляемая как

$$f(Z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{Z^2}{2}}. \quad (4)$$

На рисунке 3 представлена проверка гипотезы H_0 : статистически значимых различий связей показателей X и Y по данным выборок I и II нет.

$$z1 := \frac{1}{2} \cdot \ln \left(\frac{1+r1}{1-r1} \right) \quad z1 = -0.24 \quad \text{Преобразование Фишера для } r1$$

$$z2 := \frac{1}{2} \cdot \ln \left(\frac{1+r2}{1-r2} \right) \quad z2 = -0.282 \quad \text{Преобразование Фишера для } r2$$

$$z := \frac{|z1 - z2|}{\sqrt{\frac{1}{n1-3} + \frac{1}{n2-3}}} = 0.234$$

$$fz := 0.593 \quad \alpha := 2 \cdot (1 - fz) = 0.814$$

$$V(\alpha) := \begin{cases} y \leftarrow "H0" & \text{if } \alpha > 0.1 \\ y \leftarrow "H1" & \text{otherwise} \end{cases}$$

$$V(\alpha) = "H0" \quad \text{Статистический вывод}$$

Рис. 3. Применение Z-критерия Фишера в системе Mathcad

Как следует из рисунка 3, связи показателей X и Y по данным выборок I и II значимо не различаются.

Исследование отношений между свойствами можно выполнять в специализированных системах компьютерной математики, таких как STATISTICA, SPSS, STATGRAPHICS и других. Средства системы Mathcad также позволяют решать рассмотренные задачи, при этом использование данной системы является более прозрачным для пользователя в плане применяемого алгоритма. На практике при выполнении исследований часто эффективным является комбинирование возможностей различных программ.

Литература

1. Некрасов С.Д. Математические методы в психологии (MS Excel): учеб. пособие. 3-е изд., испр. и доп. Краснодар: Кубанский гос. ун-т, 2014. 147 с.
2. Суханова А.Г., Солоднева Д.А. Исследование эмоционального интеллекта в юношеском возрасте методами кластерного анализа // Информационные технологии в моделировании и управлении: подходы, методы, решения. V Всероссийская научная конференция с международным участием: сборник материалов: отв. за выпуск Е.В. Панюкова. Тольятти, 2022. С. 406-412.
3. Волхонская Е.В., Коротей Е.В. Моделирование случайных процессов в среде MathCAD: учебно-методическое пособие по дисциплине «Основы статистической радиотехники». Калининград: Изд-во БГАРФ, 2019. 184 с.
4. Расолько Г.А., Кремень Е.В., Кремень Ю.А. Технологии программирования: математическое моделирование и система компьютерной математики Mathcad: учеб.-метод. Пособие. Минск: БГУ, 2022. 1 электрон. опт. диск (CD-ROM). ISBN 978-985-881-167-9.

A.G. Sukhanova

Military Academy of Military Air Defense of the Armed Forces of the Russian Federation

STUDY OF RELATIONSHIPS BETWEEN PROPERTIES IN THE MATHCAD SYSTEM

Keywords: *computer modeling, Mathcad system, statistical hypothesis testing, Fisher's Z-test, correlation coefficient.*

Abstract. *This article examines the relationship between properties using Fisher's Z-test, using a specific example. Modeling of the initial data and hypothesis testing were performed using the Mathcad computer mathematics system.*

РАЗЛИЧНЫЕ ТИПЫ МОДЕЛЕЙ ДЛЯ РАЗРАБОТКИ РАДИОЭЛЕКТРОННЫХ СИСТЕМ

Ключевые слова: *радиотехническая система, математическое моделирование, имитационная модель, детерминированная модель, иерархический подход.*

В статье рассмотрены различные типы математических моделей, применяемых при разработке сложных технических систем. В зависимости от решаемых с помощью моделирования задач целесообразно использовать различные модели – от самых общих (аналитических и детерминированных) до подробных имитационных. Для уменьшения размерности сложных моделей предлагается использовать иерархический подход с заменой подробных моделей подсистем на их статистические аналоги, обобщенные данные о функционировании которых получены на соответствующих подмоделях.

Введение. В настоящее время математическое моделирование является основным инструментом, используемым при проектировании сложных технических систем. Несмотря на кажущуюся простоту использования этого метода – разработать подробную математическую модель, проверить ее на адекватность, провести серию численных экспериментов и получить требуемый результат – в ряде случаев некорректное его использование может приводить к парадоксальным результатам.

Под термином «математическое моделирование» в классической литературе понимается метод исследования объекта на основе его математической модели (математических и алгоритмических соотношений, определяющих характеристики объекта), проведение численных экспериментов и распространение результатов моделирования на реальные объекты [1]. Однако разнообразие стоящих перед разработчиками задач, а также типов моделей требует более конкретных описаний.

Постановка задачи и критерии эффективности как основа моделирования

Стандартная математическая модель сложной технической системы создается для решения триединой задачи: определить, как ведет себя

сложная система при заданных входных (начальных, возмущающих) условиях, какие входные условия надо задать, чтобы получить требуемое поведение системы и как система влияет на окружающие объекты [2]. При кажущемся разнообразии формулировок для инженерной практики они сводятся к одной – определить поведение системы в заданных условиях и при необходимости доработать ее параметры.

В зависимости от этапа разработки (начальный этап оценки технических предложений, этап выбора оптимальных параметров при уже выбранной структуре, этап натурных испытаний) изменяется цель и постановка задачи исследования, следовательно, меняется подход к принципам создания математической модели.

Для эффективного применения математического моделирования требуется корректно решить следующие вопросы.

1. Какова цель математического моделирования, то есть на какие конкретные вопросы требуется получить ответы. Например, при разработке радиолокационной системы на начальных этапах требуется определить основные характеристики – частотный диапазон, максимальную дальность работы, достижимую точность. В дальнейшем исследуются частные параметры – методы модуляции сигнала, параметры антенны, передающих и приемных устройств. На этапе испытаний системы часто возникающий вопрос – как интерпретировать поведение системы в ходе натурального эксперимента и определить возможные неисправности.

2. Какие критерии эффективности следует использовать. Интуитивно понимаемые понятия «лучше» и «хуже» нуждаются в явных формульных определениях. Так, для радиолокационной системы «лучше» может означать как «точнее», так и «легче по массе» или «меньше по объему» и так далее.

3. Какие граничные условия необходимо поставить при моделировании а также какими факторами можно пренебречь. Например, следует ли учитывать кривизну земной поверхности или разброс константы ускорения свободного падения в различных точках.

4. Чем подтверждается адекватность модели. Некорректный ответ на этот вопрос лишает смысла математическое моделирование.

Классификация математических моделей радиоэлектронных систем

При разработке радиоэлектронных систем математическое моделирование является одним из основных инструментов [3]. Можно рассматривать следующие основные типы моделей: аналитические, детерминированные, статистические и имитационные.

Аналитические модели – это модели, в которых поведение системы описывается аналитическими уравнениями, и можно считать эти модели

математическими, если для решения уравнений использованы компьютерные технологии. В реальности наличие ограничений и логических условий функционирования позволяет использовать такие модели только в ограниченных обобщенных ситуациях.

В детерминированных моделях объект описывается совокупностью математических и алгоритмических соотношений без учета случайных факторов, то есть выходные параметры модели однозначно определяются на основе ее входных параметров. Такие модели, отличаясь сравнительной простотой, позволяют определять номинальное поведение системы при штатном функционировании.

В статистических моделях учитываются недетерминированные воздействия на систему, разбросы внутренних параметров. Имеется возможность моделировать нетривиальные случайные процессы, в том числе негауссовы помехи. Метод моделирования Монте-Карло позволяет определять возможные отклонения поведения системы от номинальных, оценивать границы работоспособности системы, вероятности ошибок передачи данных и другие параметры. За возможность получения большего объема информации приходится расплачиваться значительной сложностью модели.

Имитационные модели предназначены для контроля поведения системы с течением времени. В этих моделях можно проследить внутреннее состояние в каждый момент времени, что особенно важно при создании динамических систем, например спутниковых систем связи. Сложность создания, верификации и эксплуатации имитационных моделей высока, однако их использование дает максимальную информацию о поведении системы, то есть решает основную задачу моделирования наиболее точно.

Для упрощения имитационных моделей предлагается использовать иерархический подход с сочетанием имитационных и статистических моделей [4]. Современные вычислительные средства позволяют создавать имитационные модели любой степени детализации, однако с увеличением размерности моделей многократно возрастает сложность верификации и проведения численных экспериментов. Для преодоления такого рода проблем предлагается использовать иерархический подход: в соответствии со структурой системы (разбиение системы на ряд подсистем и определение связей между ними) модель строится как совокупность отдельных подмоделей, каждая из которых описывает поведение отдельных подсистем. Изучив поведение подсистемы в различных условиях с помощью подробной имитационной модели, учитывающей множество факторов, и сформулировав обобщенные статистические характеристики подсистемы, можно заменить ее в общей модели на статистический аналог меньшей сложности. Например, в имитационной модели многоканальной системы связи можно изучить

влияние параметров распространения радиоволн при различных их отражениях, определить обобщенные точностные характеристики и использовать их статистические аналоги при моделировании системы в целом.

Примеры решения задач с помощью различных моделей

Для иллюстрации рассмотрим простейшую задачу из школьного курса физики и способы ее решения методами математического моделирования.

Задача: определить, под каким углом к горизонту надо бросить камень, чтобы он упал на максимальном расстоянии. Сразу введем ограничения: рассматривается ускорение свободного падения $g = -9,81 \text{ м/с}^2$, не учитываем кривизну поверхности (так как в этой конкретной задаче рассматривается бросок камня, при рассмотрении полета пушечного снаряда это предположение уже будет некорректно). На рисунке 1 представлены начальные условия задачи и траектория движения.

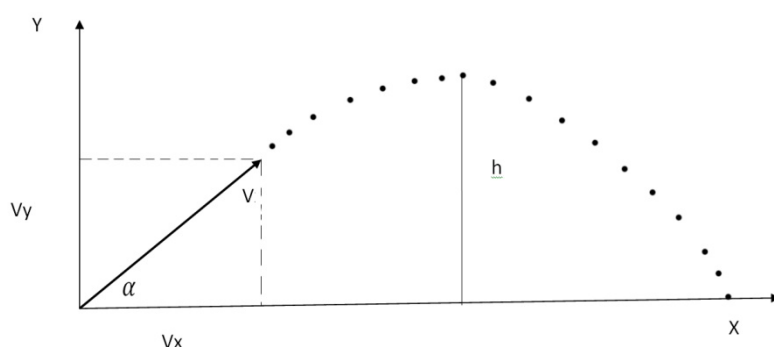


Рис. 1. Траектория движения

Определим ход решения задачи: зададим начальную скорость V и угол α броска и будем рассматривать движение в декартовой системе координат (X, Y) , где Y – вертикальная ось. Тогда составляющие скорости в начальный момент времени $t = 0$

$$V_{x0} = V \cos \alpha, V_{y0} = V \sin \alpha. \quad (1)$$

Рассмотрим два этапа полета камня. На первом этапе камень летит вверх, постепенно замедляясь до нулевой вертикальной скорости под действием ускорения g , при этом движение по горизонтали продолжается с постоянной скоростью V_{x0} . Максимальная высота определяется из условия $V_y = 0$, при этом время

$$t_1 = -\frac{V_{y0}}{g}, h = -\frac{V_{y0}^2}{2g}. \quad (2)$$

На втором этапе камень падает вниз до нулевой высоты за время

$$t_2 = \sqrt{-\frac{2h}{g}}, \quad (3)$$

продолжая лететь по горизонтали. За общее время полета $t = t_1 + t_2$ пройденное расстояние будет $S = V_{x0} \cdot t$. Далее можно составить таблицу значений расстояния S в зависимости от угла α и найти требуемое максимальное расстояние и угол. На основе представленных соотношений может быть разработана детерминированная модель, решающая поставленную задачу.

Из соотношений (1) – (3) легко получить аналитическое выражение

$$S = \frac{2V^2}{g} \sin \alpha \cdot \cos \alpha, \quad (4)$$

и его максимум достигается при $\alpha = \pi/2$.

Выражение (4) – это аналитическая модель.

Для создания имитационной модели используем метод интегрирования Эйлера для решения уравнений движения

$$\frac{dv_y}{dt} = g, \quad \frac{dy}{dt} = V_y, \quad \frac{dx}{dt} = V_x \quad (5)$$

с начальными условиями (1), причем рассмотрим отдельно первый этап достижения нулевой вертикальной скорости и второй этап достижения нулевой высоты. Получив конечное значение x , можно повторить моделирование при различных α и найти максимальное значение расстояния и соответствующий ему угол.

При решении поставленной задачи имитационная модель кажется излишне усложненным путем, детерминированная – требует составления итоговой таблицы, в которой значение угла будет найдено лишь приблизительно, а аналитическая модель решает поставленную задачу точно. Однако если усложнить условия, добавив, например, сопротивление ветра при полете, то аналитическое решение уже будет слишком сложным для явного решения, а детерминированная модель лишь немного усложнится. Если предположить, что влияние ветра может носить случайный нестационарный характер, то единственным способом решения будет добавить в имитационную модель случайные аддитивные составляющие скорости и применить метод Монте-Карло.

Для общности можно рассмотреть иерархическую модель, в которой имитируются случайные изменения ветра (то есть случайные колебания ускорений по осям), определить среднюю скорость движения и использовать это значение в детерминированной модели. Хотя такой подход при решении простейшей задачи носит только иллюстративный характер.

В следующем примере использование аналитической модели некорректно. Рассмотрим математическую модель системы связи при

использовании квадратичной амплитудной (КАМ) модуляции. Здесь сигнал

$$s(t) = I \cdot \cos 2\pi f t + Q \cdot \sin 2\pi f t \quad (6)$$

с частотой f представляет собой сумму двух ортогональных компонент и кодируется положением на комплексной плоскости его составляющих (I, Q) .

С математической точки зрения детектирование сигнала – это определение коэффициентов при преобразовании Фурье, но если в математической модели применить это стандартное преобразование, то получим явное нарушение адекватности модели. В действительности в квадратурном детекторе происходит умножение сигнала на опорные колебания $\cos 2\pi f t$ и $\sin 2\pi f t$, затем сигнал проходит через низкочастотный фильтр, и путем определения минимального отклонения при сравнении с возможными парами значений (I, Q) исходный сигнал однозначно определяется (рис. 2).

Математическая модель системы связи, определяющая вероятностные характеристики, корректно работает при использовании модели реальных детекторов, тогда как чисто аналитическое описание процесса не может учесть особенностей аппаратной реализации, например неидеальность характеристик низкочастотных фильтров.

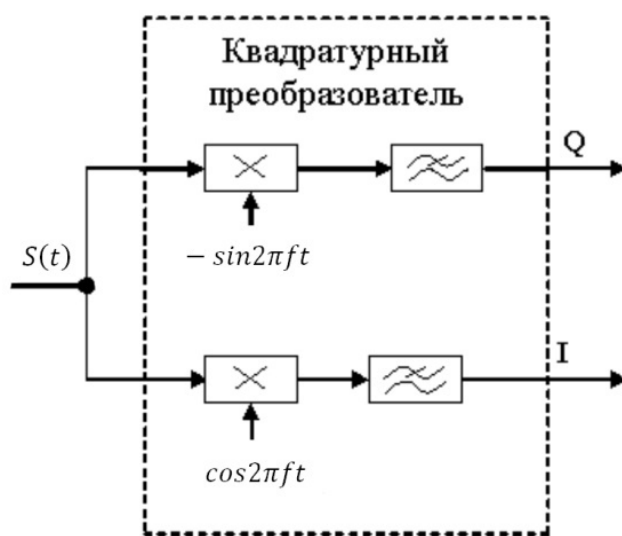


Рис. 2. Схема квадратурного детектора

При разработке комплексной модели системы связи можно использовать иерархический подход: на отдельной модели детектора определить его статистические характеристики (вероятности правильного обнаружения) в различных условиях, а затем использовать уже статистический аналог.

Заключение

Успешная разработка современных радиоэлектронных систем требует комплексного применения различных типов математических моделей. Аналитические и детерминированные модели используются на этапах начального проектирования систем, а статистические и имитационные позволяют проводить учет вероятностных и точностных характеристик поведения систем при различных типах возмущений. Иерархический подход к построению моделей может значительно уменьшить затраты на их разработку и дальнейшее применение.

Ключевым фактором успеха является правильный выбор типа модели, адекватного решаемой задаче, и чёткая постановка целей моделирования с определением критериев эффективности.

Литература

1. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. 2-е изд., испр. М.: Физматлит, 2001. 320 с.
2. Зарубин В.С. Математическое моделирование в технике / под ред. В.С. Зарубина, А.П. Крищенко. 2-е изд., стереотип. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 496 с.
3. Тихонова О.В. Комбинированные методы моделирования радиотехнических систем // Системы компьютерной математики и их приложения. Смоленск: Изд-во СмолГУ, 2014. Вып. 15. С. 56-58.
4. Тихонова О.В. Многоуровневое математическое моделирование сложных систем // Системы компьютерной математики и их приложения. Смоленск: Изд-во СмолГУ, 2012. Вып. 13. С. 216-217.

O.V. Tikhonova

Russian Technological University MIREA

DIFFERENT TYPES OF MODELS FOR THE DEVELOPMENT OF RADIO-ELECTRONIC SYSTEMS

Keywords: *radio engineering system, mathematical simulation, simulation model, deterministic model, hierarchical method.*

Abstract. *This paper examines various types of mathematical models used in the development of complex engineering systems. Depending on the tasks solved by modeling, the choice of model type depends on the specific modeling objectives – from the most general (such as analytical and deterministic) to detailed simulation models. To reduce the dimensionality of complex models, it is proposed to use a hierarchical approach, replacing detailed subsystem models with their statistical analogues, the generalized performance data of which are obtained from corresponding submodels.*

ПРОГРАММНЫЕ СРЕДСТВА ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ НЕЧЕТКОГО СИТУАЦИОННО-КОГНИТИВНОГО ПОДХОДА

Ключевые слова: *интеллектуальная поддержка принятия решений, ситуационная осведомленность, нечеткий ситуационно-когнитивный подход, мягкие ситуационно-когнитивные модели.*

В статье рассматриваются разработанные программные средства интеллектуальной поддержки принятия решений на основе нечеткого ситуационно-когнитивного подхода, включающие в себя: подсистему построения и структурно-параметрической настройки оригинальных мягких ситуационно-когнитивных моделей; подсистему интеллектуальной поддержки принятия решений; базу создаваемых мягких ситуационно-когнитивных моделей; оперативную базу данных для накопления, представления и обработки информации от датчиков и экспертов; пользовательский интерфейс для специалистов предметной области, инженеров по знаниям и лиц, принимающих решения.

Введение

Одним из активно развивающихся направлений исследований, ориентированных на расширение интеллектуальных возможностей современных информационных систем и технологий, является создание методов и средств интеллектуальной поддержки принятия решений (ИППР) в условиях неопределенности, учитывающих как ситуационную осведомленность о решаемых проблемах (проблемных ситуациях), так и выработку решений в зависимости от различных стратегий управления и сценариев достижения целевых ситуаций [1–3].

Известны различные методы и средства, которые используются для выполнения различных задач ИППР, в той или иной степени учитывающие специфику ситуационного управления в условиях неопределенности, на основе: нечетких продукционных, нейросетевых, нейро-нечетких и нечетко-нейросетевых моделей; эволюционных, генетических и биоинспирированных методов; методов роевого интеллекта; методов теории принятия решений.

Вместе с тем перечисленные выше классы методов и средства не в полной мере ориентированы на решение комплекса задач ИППР в условиях неопределенности, учитывающих ситуационную осведомленность о решаемых проблемах, выработку решений в

зависимости от различных стратегий управления и сценариев достижения целевых ситуаций.

Модульная структура и краткое описание разработанных программных средств ИППР на основе нечеткого ситуационно-когнитивного подхода

В статье рассмотрены программные средства ИППР, разработанные на основе новой разновидности мягких ситуационно-когнитивных моделей (МСКМ) [4], модульная структура которых представлена на рисунке 1 [5].

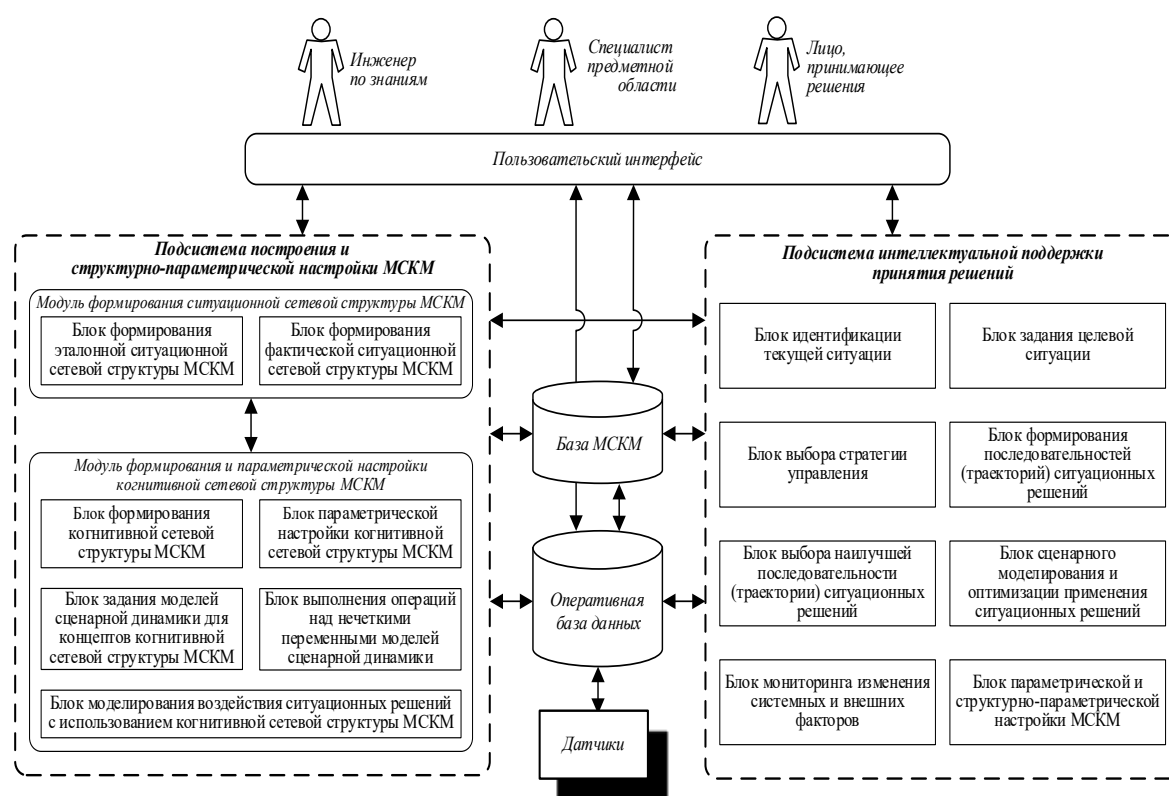


Рис. 1. Структура программных средств ИППР на основе МСКМ

Созданные программные средства включают в себя:

- подсистему построения и структурно-параметрической настройки МСКМ, состоящую:
 - из модуля формирования ситуационной сетевой структуры (включающего в себя блоки: формирования эталонной ситуационной сетевой структуры МСКМ; формирования фактической ситуационной сетевой структуры МСКМ);
 - из модуля формирования и параметрической настройки когнитивной сетевой структуры МСКМ (включающего в себя блоки формирования когнитивной сетевой структуры МСКМ; параметрической настройки когнитивной сетевой структуры МСКМ; задания моделей сценарной динамики для концептов когнитивной

сетевой структуры МСКМ; выполнения операций над нечеткими переменными моделей сценарной динамики; моделирования воздействия ситуационных решений с использованием когнитивной сетевой структуры МСКМ);

- подсистему ИППР (включающую в себя блоки идентификации текущей ситуации; задания целевой ситуации; выбора стратегии управления; формирования последовательностей (траекторий) ситуационных решений; выбора наилучшей последовательности (траектории) ситуационных решений; сценарного моделирования и оптимизации применения ситуационных решений; мониторинга изменения системных и внешних факторов; параметрической и структурно-параметрической настройки МСКМ);

- базу создаваемых МСКМ;
- оперативную базу данных для накопления, структурированного представления и обработки информации, получаемой от датчиков и предоставляемой экспертами;
- пользовательский интерфейс для специалистов предметной области, инженеров по знаниям и лиц, принимающих решения.

Разработанные программные средства позволяют осуществить построение и структурно-параметрическую настройку мягких ситуационно-когнитивных моделей, которые в отличие от известных гибридных моделей характеризуются во-первых, структурно-параметрическим соответствием между формируемыми ситуационной и двойственной к ней когнитивной сетевыми структурами МСКМ; конвергентностью и взаимообусловленностью процедур построения этих структур и их структурно-параметрической настройки в рамках единой МСКМ, обеспечивающих совместное применение ситуационной и когнитивной сетевых структур МСКМ для повышения качества принимаемых решений [6].

В то же время при построении и настройке МСКМ, в отличие от известных методов, для оценки воздействия ситуационных решений на зависимые признаки, а также для учета распределенного во времени их воздействия используются результаты когнитивного моделирования распределения воздействия ситуационных решений по когнитивной сетевой структуре МСКМ. В результате формуется фактическая ситуационная сетевая структура МСКМ, в которой зависимость ситуационных признаков учитывается за счет формирования фактических ситуаций и фактических ситуационных решений в сетевой ситуационной структуре МСКМ в результате сценарного моделирования с использованием когнитивной сетевой структуры МСКМ.

Далее разработанные программные средства позволяют с применением построенной обученной МСКМ осуществить ИППР:

- выполнить идентификацию текущей ситуации в соответствии с фактической ситуационной сетевой структурой МСКМ;
- задать целевую ситуацию из фактической ситуационной сетевой структуры МСКМ;
- выбрать стратегию управления для достижения целевых ситуаций в ситуационной сетевой структуре МСКМ;
- сформировать последовательности (траектории) ситуационных решений в ситуационной сетевой структуре МСКМ в зависимости от выбранной стратегии;
- выбрать наилучшую траекторию ситуационных решений в ситуационной сетевой структуре МСКМ для выбранной стратегии;
- выполнить сценарное моделирование и оптимизацию применения ситуационных решений из выбранной наилучшей последовательности этих решений на основе использования когнитивной сетевой структуры МСКМ;
- осуществить мониторинг изменения системных и внешних факторов;
- обеспечить параметрическую и структурно-параметрическую настройку МСКМ по результатам мониторинга.

Для тестирования параметров производительности разработанных программных средств ИППР на основе МСКМ (в том числе для оценки соотношения вычислительных и обменных операций) использовалось созданное программное обеспечение [7], адаптированное под возможности гибридного вычислительного кластера филиала НИУ «МЭИ» в г. Смоленске ([8]) с применением технологий параллельного программирования.

Заключение. Использование разработанных программных средств позволяет обеспечить повышение качества принимаемых решений в условиях неопределенности для различных стратегий управления и сценариев достижения целевых ситуаций за счет совместного использования ситуационной и когнитивной структур МСКМ, формирования наилучшей последовательности ситуационных решений, с обоснованием времени их применения при учете реакции ситуационных признаков на воздействие.

Литература

1. Kim H.J., Hong H. Predicting Information Behaviors in the COVID-19 Pandemic: Integrating the Role of Emotions and Subjective Norms into the Situational Theory of Problem Solving (STOPS) Framework // Health Communication. 2021. Vol. 37(13), P. 1640-1649.
2. Chon M.-G., Tam L., Lee H., Kim J.-N. Situational Theory of Problem Solving (STOPS). A Foundational Theory of Publics and Its Behavioral Nature in Problem Solving

/ С. Botan, E. Sommerfeldt (Eds). Public Relations Theory III: In the Age of Publics (1st ed.). Routledge. 2023. 556 p.

3. Грушо А.А., Забейло М.И., Зацаринный А.А. Обеспечение ситуационной осведомленности в интеллектуальных системах поддержки управленческих решений // Искусственный интеллект и принятие решений. 2026. № 1. С. 21-37.

4. Борисов В.В., Федулов А.С., Федулова С.А. Поддержка принятия ситуативных решений на основе сочетания нечеткого ситуационного и когнитивного подходов // Искусственный интеллект и принятие решений. 2025. № 4. С. 95-111.

5. Борисов В.В., Жарков А.П., Федулова С.А., Черновалова М.В. Программа анализа и моделирования сложных процессов на основе нечеткого ситуационно-когнитивного подхода // Свидетельство о государственной регистрации программы для ЭВМ № 2025691523 от 14.11.2025.

6. Борисов В.В., Федулов А.С., Федулова С.А. Мягкие ситуационно-когнитивные модели для интеллектуальной поддержки принятия решений // Двадцать вторая Национальная конференция по искусственному интеллекту с международным участием, КИИ-2025 (Санкт-Петербург, 6-10 октября 2025 г.). Труды конференции. В 3 т. Т. 2. С 20-32.

7. Федулова А.С., Федулов А.С., Федулов Я.А., Федулова С.А. Программа тестирования параметров производительности на гибридном вычислительном кластере // Свидетельство о государственной регистрации программы для ЭВМ № 2021613757 от 12.03.2021.

8. Борисов В.В., Зернов М.М., Федулов А.С., Якушевский К.А. Исследование характеристик гибридного вычислительного кластера // Системы управления, связи и безопасности. 2016. № 4. С. 129-146.

S.A. Fedulova

*Smolensk Branch of the National Research University
«Moscow Power Engineering Institute»*

SOFTWARE TOOLS FOR INTELLIGENT DECISION SUPPORT BASED ON A FUZZY SITUATIONAL-COGNITIVE APPROACH

Keywords: *intellectual decision support, situational awareness, fuzzy situational-cognitive approach, soft situational-cognitive models.*

Abstract. *This paper examines the developed software tools for intelligent decision support based on a fuzzy situational-cognitive approach, which include: subsystem for designing and teaching of soft situational-cognitive models; subsystem for intelligent decision support; database of Soft Situational-Cognitive Models being created; operational database for accumulating, presenting and processing information from sensors and experts; user interface for domain specialists, knowledge engineers and decision makers.*

МЕТОДИКА ОЦЕНКИ КРИТИЧНОСТИ ДЕФЕКТОВ КОРПУСОВ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ

Ключевые слова: автоматизация, корпуса интегральных микросхем, контроль качества, компьютерное зрение, поиск дефектов, критичность дефектов.

В статье представлена методика оценки критичности дефектов корпусов интегральных микросхем в системах автоматизированного контроля на основе алгоритмов компьютерного зрения и нейронных сетей. Предлагаемый подход ориентирован не только на детекцию и классификацию дефектов, но и на формирование маски их критичности с учетом типа, размеров, формы и пространственного положения относительно функционально значимых зон корпуса. Методика включает сегментацию элементов корпуса, определение параметров дефектов и построение карты риска, отражающей степень их влияния на качество изделия. Особенностью подхода является иерархическая модель оценки, позволяющая перейти от локальных характеристик дефекта к интегральной метрике качества компонента. Основные преимущества методики заключаются в повышении обоснованности принимаемых решений, снижении влияния человеческого фактора и увеличении точности оценки пригодности корпусов интегральных микросхем.

Повышение требований к надежности и качеству компонентов микроэлектроники обуславливает необходимость совершенствования методов их верификации и диагностики. Современные производственные процессы характеризуются высокой плотностью компоновки, сложностью конструктивных решений и использованием прецизионных технологий, включая лазерную обработку. В этих условиях даже незначительные дефекты корпусов интегральных микросхем могут оказывать существенное влияние на функциональные характеристики изделий. Автоматизированные системы визуального контроля, основанные на методах компьютерного зрения и машинного обучения, обеспечивают высокую скорость анализа и снижение влияния человеческого фактора. Однако большинство существующих решений ориентированы преимущественно на детекцию и классификацию дефектов без формализации их критичности. Итоговое решение о пригодности изделия

часто принимается на основе набора независимых параметров, без учета их взаимосвязи и пространственного влияния.

Таким образом, актуальной является задача разработки методики, позволяющей перейти от локальной детекции дефектов к построению интегральной оценки их влияния на качество корпуса ИМС. Центральным элементом такой методики является формирование маски критичности дефектов – количественного представления степени риска в каждой точке анализируемого изображения.

Классические методы поиска дефектов основаны на анализе локальных аномалий изображения: пороговой сегментации, морфологических преобразованиях, анализе контуров и текстурных характеристик. Такие методы эффективны при контроле простых поверхностей и хорошо структурированных объектов. Однако они обладают рядом ограничений: отсутствие учета формы и пространственной значимости дефекта, невозможность комплексной оценки влияния дефекта на функциональные зоны, изолированная оценка параметров без формирования единой метрики качества. Современные нейросетевые модели позволяют существенно повысить полноту обнаружения и точность классификации дефектов. Тем не менее в большинстве случаев результат их работы ограничивается бинарной или категориальной разметкой. Пространственное влияние дефекта за пределами его контура, а также его значимость относительно монтажной площадки или контактных областей, как правило, не формализуются.

Следовательно, возникает необходимость интеграции результатов нейросетевой детекции с моделью оценки критичности.

Методика оценки критичности дефектов

В основе предложенной методики лежит последовательная обработка изображений корпуса интегральных микросхем и объединение нейросетевых методов сегментации и детекции с формализованной моделью оценки риска.

На начальном этапе исследования осуществляется захват изображений с использованием высокоточных оптических систем, что позволяет получить детализированное представление о структуре корпуса микросхемы и создает основу для дальнейшего анализа. Далее производится обработка полученных изображений с целью выделения ключевых конструктивных элементов корпуса, включая рабочую площадку, защитное стекло и контактные выводы. Сегментация этих элементов выполняется с применением глубоких нейронных сетей, построенных на архитектуре U-Net. Использование таких моделей обеспечивает высокую точность выделения границ элементов и классификации благодаря способности автоматически извлекать сложные визуальные признаки из изображений. В частности, U-Net позволяет точно

сегментировать монтажные площадки, зоны заливки и другие важные компоненты корпуса.

С использованием сегментационных нейронных сетей строятся маски монтажной площадки, контактных зон, областей заливочного стекла и других конструктивных элементов. Эти маски определяют функционально значимые области, относительно которых в дальнейшем оценивается опасность дефектов. Таким образом создается геометрическая основа для пространственного анализа.

Применение глубоких нейросетевых моделей оправдано их устойчивостью к изменению условий освещения, геометрическим искажениям и разнообразию конструктивных особенностей микросхем. В отличие от традиционных методов обработки изображений, эти сети не требуют ручного подбора признаков и способны эффективно обучаться на больших наборах данных, что обеспечивает универсальность и стабильность работы в различных условиях.

На последующем этапе выполняется автоматизированная детекция дефектов. Система анализирует изображения корпусов микросхем, используя предварительно сформированные маски с выделенными конструктивными элементами. Маски выполняют функцию дополнительных информационных каналов, позволяя модели точнее определять границы элементов, различать допустимые отклонения и выявлять потенциальные дефекты. Комплексный анализ включает сегментацию отдельных частей корпуса, выявление аномалий и классификацию обнаруженных дефектов. Детекция охватывает широкий спектр возможных неисправностей, включая механические повреждения, деформации выводов, трещины, сколы, а также дефекты, связанные с неравномерным нанесением защитного покрытия. Кроме того, алгоритмы контроля позволяют выявлять наличие посторонних включений, способных негативно повлиять на функциональные характеристики изделия.

Для каждого выявленного дефекта вычисляются геометрические характеристики, включая площадь, периметр, линейные размеры, координаты центра масс и параметры формы. В частности, используется изопериметрическое отношение как показатель компактности формы дефекта, что позволяет учитывать его потенциальную протяженность и вероятность распространения влияния.

После получения набора параметров формируется интегральная объектная характеристика критичности дефекта. Она учитывает тип дефекта, его размеры, форму и пространственное положение относительно функциональных зон корпуса.

$$J = \sum_i \alpha_{T_i} \cdot \frac{S_i}{S_{общая}} \cdot \exp\left(-\frac{d(x)}{\lambda}\right) \cdot \frac{P_i^2}{4\pi S_i},$$

где i – индекс дефекта, J – интегральная (объектная) оценка критичности, α – вес типа дефекта, T_i – класс дефекта, S_i – площадь дефекта, S_i – площадь анализируемой области, $d(x)$ – расстояние дефекта до критического элемента корпуса, λ – параметр пространственного затухания, P_i – периметр дефекта.

Данный функционал представляет собой количественную оценку, которая может задаваться экспертно через весовые коэффициенты либо обучаться на размеченных данных. Таким образом каждому дефекту сопоставляется числовое значение, отражающее его локальную опасность.

На основе объектных оценок строится индикаторное поле дефектов – дискретная карта риска. Для каждой точки изображения определяется уровень риска в зависимости от принадлежности к области дефекта. Используется индикаторная функция множества, при которой вклад дефекта учитывается внутри его границ. При пересечении областей дефектов их вклады агрегируются по заданному правилу.

$$R_i(x) = \sum_i \alpha_{T_i} \cdot 1_{x \in D_i} \cdot \exp\left(-\frac{d(x)}{\lambda}\right),$$

где $R_i(x)$ – значение поля риска в точке x , D_i – область (множество пикселей) дефекта, $1_{x \in D_i}$ – индикаторная функция, α – вес типа дефекта, $d(x)$ – расстояние от точки до критического элемента, λ – параметр пространственного затухания.

Однако дискретная модель не отражает влияние дефекта за пределами его границ. В реальных условиях даже дефект, расположенный вблизи монтажной площадки, может оказывать влияние на пригодность изделия. Для учета пространственного воздействия формируется гладкое потенциальное поле риска. Каждый дефект рассматривается как источник риска, влияние которого убывает с расстоянием. Это реализуется с использованием сглаживающих функций или ядер затухания.

$$R_i(x) = \sum_i \alpha_{T_i} \cdot \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \cdot \exp\left(-\frac{d(x)}{\lambda}\right),$$

где $R_i(x)$ – значение непрерывного поля риска, c – центр масс дефекта, σ – характерный радиус влияния дефекта, α – вес типа дефекта, $d(x)$ – расстояние до критического элемента корпуса, λ – коэффициент пространственного затухания.

Полученное непрерывное поле моделирует распространение критичности за пределы контура дефекта и позволяет учитывать перекрытие зон влияния нескольких дефектов. В результате

формируется пространственная распределенная маска критичности дефектов, в которой каждой точке корпуса сопоставляется количественная оценка риска.

Особое внимание в методике уделяется анализу монтажной площадки. После сегментации вычисляется выпуклая оболочка допустимой зоны размещения компонентов. Проверяется возможность размещения прямоугольника заданных размеров внутри доступной области. При этом карта риска накладывается на монтажную площадку, что позволяет учитывать не только геометрическую пригодность, но и распределение критичности внутри допустимой зоны. Таким образом объединяются геометрический и риск-ориентированный анализ.

Финальным этапом является формирование интегральной оценки качества корпуса ИМС. Она может рассчитываться на основе максимального значения риска, среднего уровня риска по функциональным зонам, площади областей повышенной критичности и соблюдения требований к доступной зоне монтажной площадки. Полученная метрика используется для автоматизированного принятия решения о пригодности изделия.

Предложенная методика оценки критичности дефектов обеспечивает переход от бинарной логики «дефект обнаружен» к количественной модели пространственного риска. В отличие от традиционных подходов, она учитывает форму и геометрию дефекта, его положение относительно функциональных зон, а также пространственное распространение влияния. Формирование маски критичности позволяет получить единую интегральную метрику качества, снизить влияние субъективного фактора и повысить обоснованность решений в системах автоматизированной верификации компонентов микроэлектроники. Возможность адаптации весовых коэффициентов и обучения модели по данным делает методику масштабируемой и применимой в различных технологических условиях.

Разработана методика оценки критичности дефектов корпусов интегральных микросхем, основанная на интеграции нейросетевой детекции и формализованной модели построения масок риска. Ключевой особенностью подхода является формирование пространственной распределенной маски критичности, отражающей не только локальные характеристики дефектов, но и их влияние на функциональные зоны корпуса. Реализация предложенной методики позволяет повысить полноту и точность оценки качества корпусов ИМС, обеспечить формализацию процесса принятия решений и создать основу для дальнейшего развития интеллектуальных систем автоматизированного контроля в микроэлектронике.

Литература

1. Гагарина Л.Г. Алгоритмы и структуры данных. М.: Финансы и статистика, 2009. 482 с.
2. Портнов Е.М., Юдахин Ю.Ю., Кокин В.В. Методика автоматизированного визуального контроля качества корпусов интегральных микросхем // Системы компьютерной математики и их приложения: межвузовский сборник научных трудов. Смоленск: Изд-во СмолГУ, 2025. Вып. 26. С. 157-162.
3. Юдахин Ю.А., Юдахин Ю.Ю., Канарейкина Д.О. Технология изготовления многовыводных рамок микросхем с применением прецизионной лазерной резки // Технологии в электронной промышленности. 2025. № 3 (144) С. 35-39.
4. Либрман А.И. Решение задачи классификации объектов на изображении методами компьютерного зрения в различных сферах человеческой деятельности // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2024. № 3. С. 74–91.
5. Полковникова Н.А. Исследование методов и алгоритмов компьютерного зрения на основе свёрточных и рекуррентных нейронных сетей // Эксплуатация морского транспорта. 2020. № 3 (96). С. 154–168.
6. Корчагин В.Д. Поиск дефектов в сварных металлических соединениях при помощи технологий компьютерного зрения // Новые материалы, оборудование и технологии в промышленности: материалы Международной научно-технической конференции молодых ученых. Могилев, 2024. С. 197–198.

Y.Y. Yudahin

National Research University of Electronic Technology

METHODOLOGY FOR ASSESSING THE CRITICALITY OF DEFECTS IN INTEGRATED CIRCUIT PACKAGES IN AUTOMATED INSPECTION SYSTEMS

Keywords: *automation, integrated circuit packages, quality control, computer vision, defect detection, defect criticality.*

Abstract. *The paper presents a methodology for assessing the criticality of defects in integrated circuit packages within automated inspection systems, based on computer vision algorithms and neural networks. The proposed approach focuses not only on the detection and classification of defects but also on generating a criticality mask that takes into account the defect type, size, shape, and spatial position relative to the functionally significant areas of the package. The methodology includes segmentation of package elements, determination of defect parameters, and construction of a risk map reflecting the extent of their impact on product quality. A key feature of the approach is a hierarchical evaluation model that allows transitioning from local defect characteristics to an overall quality metric for the component. The main advantages of the methodology are the increased justification of decision-making, reduced influence of human factors, and improved accuracy in assessing the suitability of integrated circuit packages.*

РАЗДЕЛ 2

Параллельные вычисления и многоядерные процессоры

Г.О. Абрамов, Д.С. Медведев, В.И. Мунерман, Д.А. Попов
Смоленский государственный университет

УДК 004.272

АРХИТЕКТУРА USB-КЛАСТЕРА ДЛЯ ПАРАЛЛЕЛЬНОГО УМНОЖЕНИЯ МАТРИЦ ПО АЛГОРИТМУ КЭННОНА

Ключевые слова: *параллельные вычисления, алгоритм Кэннона, кластерная система, тороидальная топология, блочное умножение матриц.*

В статье приводится теоретическая модель вычислительной системы для параллельного умножения матриц на основе алгоритма Кэннона. Под кластером в контексте данного исследования понимается совокупность вычислительных узлов, объединенных высокоскоростным каналом передачи данных с целью решения ресурсоемких задач. Предложенная модель может быть обобщена и на другие классы параллельных алгоритмов, демонстрирующих схожие шаблоны обмена данными.

Введение. В качестве программного взаимодействия рассматривается интерфейс USB4, обеспечивающий пропускную способность до 40 Гбит/с при низких программных накладных расходах и не требующий узкоспециализированных портов, в отличие от таких решений, как Ethernet 40GbE. Проведён сравнительный анализ интерфейсов USB4, Ethernet и InfiniBand (табл. 1), обоснован выбор USB4 для задач маломасштабных кластерных конфигураций. Разработана математическая модель балансировки вычислительной нагрузки, устанавливающая соотношение между производительностью вычислителя, пропускной способностью канала и критическим размером блока. Предложена топологическая организация процессорных элементов в виде тороидальных слоёв, адаптирующая идеи алгебраических моделей массовой обработки данных к классическому алгоритму Кэннона.

1. Выбор способа соединения компьютеров. В качестве способа соединения компьютеров в кластер был выбран стандарт USB4, предоставляющий канал связи пропускной способностью до 40 Гбит/с, но при этом не требующий узкоспециализированных портов, которые

нужны для других вариантов, таких как Ethernet 40GbE. Ниже приведена простая сравнительная таблица факторов, повлиявших на выбор.

Таблица 1

Характеристика	USB4 [1]	Ethernet	InfiniBand [2]
Скорость	40 – 80 Гбит/с	1(Встроен в большинство ПК) – 100 (Дискретный адаптер) Гбит/с	40 – 400 Гбит/с
Задержка	10 – 30 мкс	20 – 100 мкс	< 1 мкс
Цена адаптера	0	0 – 180 000 р	50 000 – 200 000
Максимальная длина	До 3м	До 100 км	До 10 км

Несмотря на выдающиеся технические характеристики, такие как крайне низкая латентность (менее 1 мкс) и высокая масштабируемость, от использования InfiniBand в рамках данной модели было решено отказаться. Ключевым фактором является несоответствие архитектурного подхода: InfiniBand ориентирован на создание крупных высокопроизводительных кластеров (HPC) с централизованным управлением и сложной топологией, требующей специализированного оборудования – коммутаторов, адаптеров (HCA) и кабелей. Для предложенной конфигурации, которая предполагает соединение ограниченного числа вычислительных узлов (либо связки «хост – ускоритель»), использование InfiniBand является избыточным как по стоимости, так и по сложности развертывания. Кроме того, ориентация на USB4 обусловлена стремлением к минимизации программных накладных расходов: в отличие от InfiniBand, работа с которым требует стека драйверов и промежуточного ПО, прямое управление передачей данных через USB дает большую гибкость на уровне приложения, что критично для тонкой оптимизации блочных алгоритмов.

В случае Ethernet уже существует большое количество программного обеспечения, у которого есть существенные ограничения в контексте нашей цели быстрого умножения матриц. В частности, существенным препятствием могла бы стать сетевая маршрутизация: протоколы TCP/IP вносят непредсказуемые задержки и требуют фрагментации данных. В случае USB ограничений будет меньше, что откроет новые пути оптимизации на уровне прямого управления передачами.

Использование интерфейса USB в контексте реализации алгоритма блочного умножения матриц представляет собой специфический сценарий организации вычислений. В такой конфигурации USB выполняет функцию канала передачи данных между хостом и внешним устройством.

На практике могут использоваться внешние графические процессоры, подключаемые через Thunderbolt, где USB-интерфейс обеспечивает передачу блоков матриц в видеопамять. Также возможны варианты с FPGA-платами, имеющими USB-интерфейс и реализующими алгоритм умножения на аппаратном уровне, либо с микроконтроллерами и DSP для работы с матрицами малых размеров.

2. Параллельный алгоритм умножения матриц. Применение блочного алгоритма умножения матриц (алгоритм Кэннона [3]) в условиях ограниченной пропускной способности USB обусловлено необходимостью минимизации количества пересылок данных. Вместо передачи полных матриц производится обмен блоками, что позволяет организовать конвейерную обработку: пока ускоритель вычисляет текущие блоки, шина подгружает следующие. Основной проблемой является высокая латентность USB по сравнению с шинами PCIe или InfiniBand.

Основное техническое ограничение USB – пропускная способность, которая ниже, чем у внутренних интерфейсов (память, PCIe). Для передачи больших объемов данных с контролем целостности применяется класс Bulk Transfers. В периферийных вычислениях это используется для подключения тензорных ускорителей (например, Google Coral или Intel Movidius), выполняющих задачи компьютерного зрения.

3. Выбор размерности блоков матрицы. Для оценки эффективности построим математическую модель. Рассмотрим умножение двух квадратных матриц размера n , разбитых на блоки размера b . Введем параметры: S – пропускная способность USB, P – производительность ускорителя, T – размер машинного слова. Время передачи данных для одного шага составит:

$$\frac{(2b^2T)}{S}.$$

Время вычислений:

$$\frac{(2b^3)}{P}.$$

Условие отсутствия простоев ускорителя требует выполнения неравенства

$$\frac{(2b^2T)}{S} \geq \frac{(2b^3)}{P},$$

откуда получаем, что критическое значение размера блока

$$b \geq \frac{PT}{S}.$$

Производительность ускорителя требует увеличения размера блоков, что диктует и низкая пропускная способность USB. Верхняя

граница – объем локальной памяти. Выведенное соотношение задает минимальный размер блока для исключения простоев, но это условие необходимое, но не достаточное: оно не учитывает топологию связей в многопроцессорной системе, где время коммуникации зависит от структуры связей (число пересылок и конкуренция).

4. Архитектура кластера. Сформулированные требования к минимизации простоев и эффективной загрузке вычислителей приводят к необходимости выбора такой топологии соединений, которая обеспечивает равномерную циркуляцию блоков между процессорами для обеспечения передачи блоков по кластеру без лишних издержек путем циклического смещения, что следует из условий выполнения алгоритма Кэннона. Исходная архитектура кластера приведена на рисунке 1.

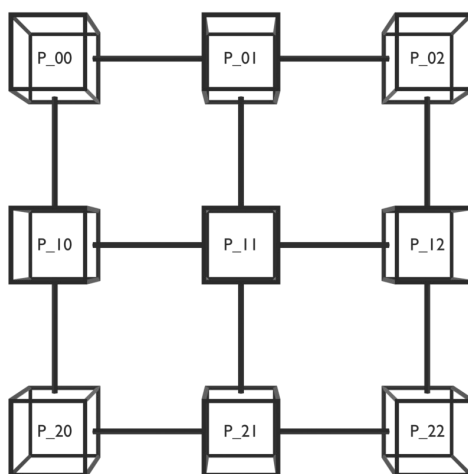


Рис. 1. Исходная архитектура кластера

Для обеспечения циклического сдвига блоков между процессорами один слой кластера, как по вертикали, так и по горизонтали, должен иметь кольцевую архитектуру (рис. 2).

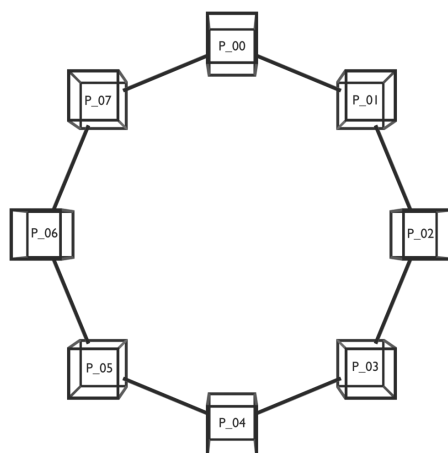


Рис. 2. Пример кольцевой архитектуры слоя кластера

Этим требованиям удовлетворяет тороидальная структура связей [4], показанная на рисунке 3, позволяющая замкнуть крайние элементы процессорной решётки и тем самым сократить путь передачи данных.

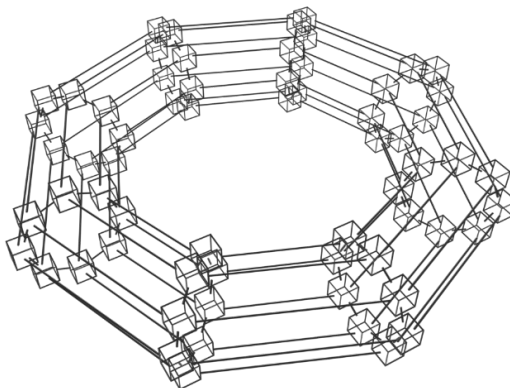


Рис. 3. Тороидальная структура кластера для умножения матриц

Организация вычислительной системы в виде двумерного тора, разбитого на горизонтальные и вертикальные кольцевые слои, обеспечивает масштабирование предложенной модели на случай больших размерностей без потери эффективности и удовлетворяет выведенному нами неравенству, а межуровневые связи реализуются по кратчайшим маршрутам замкнутого тора. Особенность алгоритма Кэннона, требующая хранения исходных матриц, операндов и результата на одном компьютере, может быть обеспечена архитектурой многопроцессорной вычислительной системы с общим управлением [5]. Это может быть отдельная вычислительная система (компьютер), выполняющий функцию хост-машины, или же один из процессоров тороида делается бифункциональным, выполняя как функцию общего управления на первом и последнем шагах (разделение матриц операндов и сборка результата), так и функции вычислительного процессора.

Заключение. Таким образом, в работе получены следующие результаты.

1. Проведенный анализ показал, что построение USB-кластера обеспечивает высокоэффективную реализацию параллельного умножения матриц, по скорости передачи данных близкую к высокоскоростным сетям, но требующую значительно меньше затрат.

2. Произведен выбор алгоритма параллельного умножения матриц, обеспечивающего естественный параллелизм.

3. Предложена тороидальная архитектура USB кластера.

Литература

1. USB4® Specification v2.0 / USB Implementers Forum. 2024. Mode of access: <https://www.usb.org/document-library/usb4r-specification-v20> (date of access: 27.02.2026).

2. InfiniBand™ Architecture Specification. Frequently Asked Questions // InfiniBand Trade Association. 2025. Mode of access: URL: <https://www.infinibandta.org/ibta-specification/> (date of access: 27.02.2026).

3. Cannon L.E. A cellular computer to implement the Kalman Filter Algorithm: Ph. D. thesis. Bozeman: Montana State University, 1969. 188 p.

4. Мунерман В.И. Массовая обработка данных. Алгебраические модели и методы: монография. М.: ИНФРА-М, 2023. 229 с.

5. Прангишвили И.В., Виленкин С.Я., Медведев И.Л. Параллельные вычислительные системы с общим управлением. Энергоатомиздат, 1983. 312 с.

G.O. Abramov, D.S. Medvedev, V.I. Munerman, D.A. Popov
Smolensk State University

USB CLUSTER ARCHITECTURE FOR PARALLEL MATRIX MULTIPLICATION USING THE CANNON ALGORITHM

Keywords: *parallel computing, Cannon's algorithm, cluster system, toroidal topology, block multiplication of matrices.*

Abstract. *This paper presents a theoretical model of a computing system for parallel multiplication of matrices based on Cannon's algorithm. In the context of this study, a cluster is understood as a set of computing nodes united by a high-speed data transmission channel in order to solve resource-intensive problems. The proposed model can be generalized to other classes of parallel algorithms that demonstrate similar patterns of data exchange.*

М.А. Акимов, С.В. Козлов
Смоленский государственный университет

УДК 004.75

СИСТЕМА РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ С ДИНАМИЧЕСКОЙ ЗАГРУЗКОЙ ЗАДАЧ

Ключевые слова: *распределённые вычисления, параллелизм, динамическая загрузка задач, клиент-серверная система, масштабируемость.*

В статье рассматривается система распределённых вычислений по модели «мастер - подчинённый» с возможностью динамической загрузки задач через внешние модули, а также аналоги системы. Такой подход обеспечивает гибкость, модульность, расширяемость системы и удобство для пользователя, располагает к созданию

общедоступной базы с уже созданными задачами. Данный способ взаимодействия системы с решаемой задачей позволяет менять её в самой программе, не перезапуская и неперекompилируя. Особое внимание уделяется анализу существующих аналогов распределённых систем и сравнению предлагаемого подхода с традиционными архитектурами, такими как MPI-кластеры.

Распределённые вычисления решают ресурсоёмкие задачи с помощью множества компьютеров. При этом используется модель «мастер - подчинённый», а именно «мастер» – это сервер, а «подчинённый» – клиент. Задача делится на части, распределяется по узлам, а после обработки результаты объединяются [1, 2]. Сеть компьютеров функционирует как единый мощный ресурс [3, 4]. Параллельное выполнение сокращает общее время решения задач, ограничиваясь временем самой продолжительной подзадачи [5].

В «мастер-подчинённой» модели сервер делит данные на части и раздаёт их клиентам. После обработки сервер собирает и объединяет результат в итоговый [6; 7]. В модели MapReduce (Hadoop) «мастер» разбивает данные на фрагменты, раздаёт их узлам для обработки, а затем на фазе Reduce собирает и объединяет результаты. Узлы системы автономны, каждый с собственной памятью, и обмениваются сообщениями. Кроме того, важны отказоустойчивость и масштабируемость [8; 9].

Существуют различные подходы и архитектуры распределённых систем [10]. В настоящее время обычно выделяют несколько типов. Так, Peer-to-Server – это простая модель, где программист вручную организует обмен данными между узлами (сокет, MPI и т.д.). В таких системах часто применяются технологии удалённого вызова процедур (RPC, DCOM, CORBA) или низкоуровневые библиотеки обмена сообщениями (MPI) для связи узлов. MPI, например, задаёт стандарт передачи сообщений между процессами: все процессы работают в собственном адресном пространстве, без общих переменных и взаимодействуют явным обменом сообщений.

Системы Volunteer/Grid-системы – это платформы для конкретных проектов (SETI@Home, BOINC). Узлы получают задания от сервера, выполняют их и возвращают результаты, по принципу «получил задачу – выполнил – вернул результат». Изначально проект ориентируется на одну задачу или семейство задач, а инфраструктура обеспечивает повторное использование вычислительных ресурсов. Клиент получает фрагмент данных, выполняет нужный расчёт и возвращает результат серверу. В таких системах координация между клиентами обычно минимальна: каждый узел работает независимо, по принципу «выбери задание – вычисли – верни».

P2P-сети – равноправные узлы без центрального сервера (BitTorrent, Bitcoin). Каждый узел может одновременно получать и передавать данные другим узлам по мере необходимости.

Центральное управление – один главный координатор управляет очередью заданий и распределением их по узлам. Это характерно для кластеров и облачных платформ (например, MPI-кластеры со службой планирования, облачные эластичные кластеры).

Эти архитектуры могут комбинироваться. Так, например, MapReduce использует централизованного «мастера» для управления заданиями и распределения данных, а клиентские узлы обмениваются результатами через распределённую файловую систему. Сетевые технологии (RPC/UDP/TCP) и Message Queue, например, RabbitMQ, Kafka, также часто выступают базовым транспортом для передачи сообщений и данных между компонентами распределённой системы.

Рассмотрим более подробно подход с динамической загрузкой задач. В этой системе вычислительная логика задаётся двумя внешними C#-модулями `ServerTask.cs` и `ClientTask.cs`, которые при запуске компилируются и загружаются в .Net Reflection. Благодаря этому платформа универсальна: чтобы добавить новую задачу, достаточно создать соответствующие файлы.

В файле `ServerTask.cs` задаётся интерфейс задачи, например `Itask` с методами разбиения входных данных на части и объединения результатов. В файле `ClientTask.cs` реализуется метод обработки одной порции данных: клиент получает фрагмент, выполняет вычисления и возвращает результат серверу.

Файл `ClientTask.cs` содержит реализацию клиентской части. В нём описывается класс задачи, например `ClientTask`, реализующий метод обработки одной порции данных. Получив свой фрагмент от сервера, клиент выполняет вычисления, как того требуют условия задачи, и возвращает результат своей порции данных обратно на сервер.

При запуске сервер компилирует файлы `ServerTask.cs` и `ClientTask.cs` через Reflection. Он разбивает входные данные на порции, отправляет их клиентам по сети или через файловую систему, а после вычислений собирает результаты вызовом `MergeResults`. Алгоритмы разбиения и объединения данных задаются внешними скриптами и не встроены в код сервера или клиента.

Такой подход обеспечивает высокую модульность. Основное приложение остаётся универсальным и лишь координирует процесс. Добавление новой задачи требует только создания соответствующих C#-скриптов. Так, скрипт Reflection позволяет динамически загружать нужные сборки и типы, как в плагинной системе. Нет необходимости перекомпилировать сервер или клиент – достаточно заменить файлы с описанием задачи.

Рассмотрим другие системы. MPI и кластеры: в классической системе MPI-приложения разработчик заранее прописывает логику разбиения данных (через вызовы MPI_Send/MPI_Recv, MPI_Scatter, MPI_Gather и т.д.). Все процессы запускаются одновременно, например с помощью `mpirun`, и каждый выполняет определённую часть кода. Процессы работают в отдельных адресных пространствах, общих переменных нет; передача данных идёт через сообщения по сети. В таком сценарии код задачи обычно статичен и скомпилирован на момент запуска кластерной работы. В отличие от этого, в описанной системе разбиение и объединение логики вынесено в загружаемые классы, а обмен данными с клиентами может осуществляться любым способом (также через сети или файловые хранилища).

Модель MapReduce / Hadoop: пользователь задаёт функции Map и Reduce (обычно на Java), а фреймворк Hadoop разбивает большой набор данных на блоки, распределяет их на узлы и собирает результаты автоматически. Код вычислений становится частью пакета (JAR), разворачиваемого на кластере. В описываемой системе аналогично есть разделение на «map» - и «редьюс» - этапы, но программы сервера и клиента универсальны, а конкретная логика настраивается файлами `.cs`. При этом сервер вручную запускает клиентов (десктопные приложения), а не использует встроенную инфраструктуру распределённого кластера Hadoop.

Волонтёрские проекты (BOINC, SETI@Home): такие системы централизованно создают проекты, в рамках которых клиент скачивает от сервера часть задачи, например, фрагмент сигнала космического телескопа или диапазон чисел, обрабатывает и отправляет ответ. При этом задача уже сформулирована сервером заранее, и клиенты выполняют строго заданную обработку. В нашей модели динамически можно менять задачи.

Сетевые сервисы (RPC, WCF, REST): часто распределённые приложения строят вокруг сервисов. Сервер публикует набор API-вызовов, например WCF-служба или REST-сервис, а клиенты отправляют запросы с параметрами, получая ответ. Такая задача обычно фиксирована на стороне сервера, как набор методов. В рассматриваемом подходе идея схожа: клиенты и сервер обмениваются сообщениями. Но отличие в том, что «методы» задачи задаются внешними скриптами. Сервер и клиент используют один и тот же способ передачи данных, однако вместо того, чтобы вызывать заранее скомпилированный метод, они динамически обращаются к методам интерфейса, описанного во входных файлах.

Таким образом, уникальность предлагаемой системы состоит в разделении вычислительной логики и механизма её запуска. Главная особенность – универсальность: одна и та же база сервера и клиентов

может выполнять любой набор задач, достаточно лишь заменить файлы .cs. При этом интерфейс взаимодействия остаётся неизменным и может быть реализован любой удобной технологией связи (сокеты, HTTP-запросы, файловый обмен и т.д.).

Распределённые вычисления нацелены на эффективную обработку тяжёлых задач посредством распараллеливания и использования ресурсов многих машин. В предложенной системе реализован нетипичный подход: вычислительная задача задаётся не в коде ядра, а как два внешних C#-модуля, динамически загружаемых приложением через скрипт Reflection. Сервер разделяет исходные данные между клиентами и собирает ответы – всю логику разбиения / склейки определяют скрипты задачи. Таким образом, рассматриваемая система распределённых вычислений демонстрирует альтернативный подход. В ней реализованы базовые принципы масштабирования и отказоустойчивости распределённых систем, при этом она остаётся максимально универсальной к виду решаемых задач.

Литература

1. Вяткин В.В., Литвинов В.В. Распределённые вычислительные системы. М.: МГТУ им. Н.Э. Баумана, 2019. 412 с.
2. Каляев И.А., Мельник Е.В., Орда-Жигулина М.В. Исследования мультипроцессорных вычислительных систем и систем управления // Наука Юга России. 2022. Т. 18, № 4. С. 85-96.
3. Григорьев П.А. Системы параллельной обработки данных. М.: Лань, 2017. 288 с.
4. Поляков А.А. Распределённые информационные системы. СПб.: Питер, 2020. 350 с.
5. Модель анализа надежности распределенных вычислительных систем / Р.Ю. Царев, А.Н. Пупков, М.А. Огнерубова [и др.] // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2013. № 1(47). С. 86-91.
6. Сахаров А.П. Высокопроизводительные вычисления: учебное пособие. Новосибирск: НГУ, 2019. 198 с.
7. Чернышев Б.А. Распределённые системы и облачные технологии. М.: НИУ ВШЭ, 2021. 240 с.
8. Гаврилов И.А., Лайко Ю.Г. Архитектуры вычислительных систем. М.: Физматлит, 2016. 304 с.
9. Ермаков С.М. Введение в распределённые вычисления. М.: URSS, 2020. 224 с.
10. Быков А.А., Козлов С.В., Морозов С.А. Проектирование архитектуры рекомендательной системы реального времени средствами векторных баз данных // Естественные и технические науки. 2023. № 11(186). С. 272-275.

M.A. Akimov, S.V. Kozlov
Smolensk State University

DISTRIBUTED COMPUTING SYSTEM WITH DYNAMIC TASK LOADING

Keywords: *distributed computing, parallelism, dynamic task loading, client-server system, scalability.*

Abstract. *This article discusses a distributed computing system based on a master-slave model with the ability to dynamically load tasks through external modules and similar systems. This approach ensures flexibility, modularity, and extensibility of the system, as well as user convenience, and facilitates the creation of a publicly accessible database of pre-defined tasks. This method of system interaction with the task being solved allows for modifications within the program itself, without restarting or recompiling. Particular attention is given to the analysis of existing distributed system analogs and a comparison of the proposed approach with traditional architectures, such as MPI clusters.*

И.С. Аникин

Национальный исследовательский университет «МИЭТ»

УДК 004.6:004.75:004.02

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ СЕТЕВОГО ТРАФИКА

Ключевые слова: *большие данные, сетевой трафик, телеметрия, событийная модель, оконная агрегация, водяные метки (watermarks), опоздавшие события, табличные и графовые признаки, граф-сигнал, энтропия Шеннона; нормализация по ключам и многовидовое представление.*

В статье рассматривается разработка математической модели представления данных сетевого трафика, предназначенной для последующего применения в задачах интеллектуального анализа и выявления скрытых аномалий. Основу модели составляют событийный подход, оконная агрегация и механизм работы с табличными и графовыми признаками. Отдельное внимание уделено обработке опоздавших событий, обеспечению обработки каждого события только один раз (Exactly-Once) и адаптации к дрейфу данных.

Актуальность разработки математической модели представления данных применительно сетевого трафика связана с рядом особенностей в данной области.

Современные сети передачи генерируют колоссальные объёмы телеметрии [1; 2]. Потоки данных NetFlow и IPFIX [3], показатели производительности по протоколу SNMP, текстовые сообщения об ошибках и событиях из журналов Syslog и т.п. Большие объёмы и разнородность данных делает задачи их формализованного представления и последующего анализа методологически и технологически сложными.

Еще одной проблемой является отсутствие единого математического аппарата, способного одновременно и с равной эффективностью описывать как табличные (атрибутивные) данные, так и графовые (топологические) взаимосвязи в сети [4-6].

Вышеупомянутые ограничения напрямую затрудняют решение одной из самых сложных практических задач — надежного обнаружения скрытых аномалий. Стандартные методы, не учитывающие многомерность и взаимосвязи в данных, часто оказываются не способными к выявлению таких угроз [7; 8].

Таким образом, разработанная математическая модель представления данных может использоваться в качестве основы для последующего построения более точных и надежных алгоритмов для выявления аномалий, а также последующей проверки и подтверждения их достоверности.

1. Событийная модель, ключи агрегации и матрица признаков

Рассматривается поток телеметрических событий:

$$\mathcal{E} = \{e_i = (\tau_i, u_i)\}_{i=1}^{\infty}, \quad (1)$$

где τ_i — метка *времени события* (event time), а u_i — вектор атрибутов (сырые поля NetFlow/IPFIX, SNMP-счетчики, Syslog-поля и т.п.) [1-3].

Для управляемого объекта (устройство, интерфейс, IP-пара, VLAN, подсеть и др.) определяем *ключ* $k \in K$.

$$W_{\Delta}[\mathcal{E}](t, k) = \text{Agg}(\{u_i: e_i = (\tau_i, u_i), \tau_i \in [t - \Delta, t), k(u_i) = k\}), \quad (2)$$

где $\text{Agg}(\cdot)$ — семейство детерминированных агрегирующих функционалов (суммы, счётчики, средние, энтропии, и пр.), а $k(u)$ — отображение атрибутов в ключ.

Унифицированный вектор признаков для ключа k в момент t :

$$x_{k, \Delta}(t) = \Phi(W_{\Delta}[\mathcal{E}](t, k)) \in \mathbb{R}^{D_{\Delta}}, \quad (3)$$

где $\Phi(\cdot)$ — оператор извлечения признаков из агрегатов. При множестве окон $\Delta = \{\Delta_s, \Delta_m, \Delta_l\}$ формируется мультишкальная матрица/вектор.

2. Водяная метка и опоздавшие события

Введем монотонно возрастающую *водяную метку* (watermark) $w(t)$, определяющую максимальную временную отметку, для которой гарантируется, что все события с меньшими метками уже получены и обработаны [1; 2]:

$$\frac{dw(t)}{dt} \geq 0, w(t) \leq t, \quad (4)$$

Событие $e = (\tau, u)$ относится к окну $[t-\Delta, t)$, если $\tau \in [t-\Delta, t)$ и $\tau \leq w(t) + \varepsilon$, где $\varepsilon \geq 0$ — допускаемая задержка для опоздавших событий.

Если $\tau > w(t) + \varepsilon$, событие является «слишком поздним» и обрабатывается особой веткой корректировок.

3. Оконная агрегация

Выбираем три характерные шкалы $\Delta_s \ll \Delta_m \ll \Delta_l$ (например, 1–5–60 мин) для фиксации краткосрочных всплесков, рабочей динамики и медленных изменений. Определим конкатенацию признаков:

$$x_k(t) = [x_{k,\Delta_s}(t) \parallel x_{k,\Delta_m}(t) \parallel x_{k,\Delta_l}(t)] \in \mathbb{R}^D \quad (5)$$

4. Библиотека признаков (табличные и графовые)

Интенсивности и доли. Для количества событий N , байтов B , пакетов P и подмножеств направлений/портов:

$$\lambda_N = \frac{N}{\Delta}, \lambda_B = \frac{B}{\Delta}, \lambda_P = \frac{P}{\Delta}, r_g = \frac{N_g}{N}, r_p = \frac{N_p}{N}, \quad (6)$$

где N_g — число событий класса g (например, по приложению), N_p — число событий для порта p ; λ — показатели характеризуют плотность потока; r — показатели — структуру трафика.

Энтропия. Для количественной оценки разнообразия трафика используем метрику энтропии Шеннона:

$$H = -\sum_j p_j \log p_j, \quad p_j = \frac{N_j}{\sum_m N_m}. \quad (7)$$

Для анализа динамики временного ряда $y(t)$ (например, объема трафика $B(t)$) в последовательные временные интервалы рассчитываются два ключевых признака, чувствительных к резким изменениям:

– **локальный градиент** $\nabla y(t) = y(t) - y(t - \Delta)$, который показывает абсолютное изменение значения метрики между текущим (t) и предыдущим ($t-\Delta$) окнами. Эта простая разность непосредственно

фиксирует мгновенные скачки или падения величины y . Резкий всплеск (burst) проявляется здесь как большое положительное значение $\nabla y(t)$;

– коэффициент вариации $cv(y) = \frac{\sigma(y)}{\mu(y)}$, показывающий, насколько

сильно значения y отклоняются от своего среднего уровня в относительных единицах. Здесь $\mu(y)$ – это скользящее среднее (характерный «фоновый» уровень), а $\sigma(y)$ – скользящее стандартное отклонение (мера разброса).

Графовые признаки. Рассматривается ориентированный (или неориентированный) граф взаимодействий в окне $[t - \Delta t, t]: G_{k,\Delta}(t) = (V, E)$. Для узла $v \in V$ (например, IP-адрес/узел сети) – $\deg(v)$, $C_{loc}(v)$, ρ_{ego} , $BC(v)$, $CC(v)$, где $\deg$ – степень узла, C_{loc} – локальный коэффициент кластеризации, ρ_{ego} – плотность эго-графа, BC/CC – посредническая/близостная центральности [4-6].

5. Проекция, нормализация и многовидовое представление

Исходные данные представлены в пространстве очень большой размерности D . Это может приводить к проблемам с вычислительной эффективностью и качеством моделей. Для решения этой проблемы используется линейное отображение $\Psi: \mathbb{R}^D \rightarrow \mathbb{R}^d$, которое преобразует вектор $x_k(t)$ в новый вектор $z_k(t)$ в пространстве гораздо меньшей размерности d .

Данные поступают от разных ключей k (например, от разных пользователей, устройств или сетевых интерфейсов). Для каждого ключа k применяем масштабирующее преобразование:

$$\tilde{x}_k(t) = S_k^{-1}(x_k(t) - \mu_k), \quad (8)$$

где μ_k – оценка среднего значения, S_k – оценка масштаба. Это устраняет межключевые различия и оцениваются не абсолютные значения, а относительные отклонения от собственной нормы.

Так как данные поступают из различных источников и могут иметь разную природу, необходимо объединить их в один общий вектор признаков x .

Пусть имеются табличные признаки x^{tab} и графовые x^g . Тогда

$$x = [\alpha x^{tab} \parallel (1 - \alpha)x^g], \alpha \in [0, 1] \quad (9)$$

Таким образом, разработанная математическая модель представления данных сетевого трафика объединяет оконную агрегацию, графовый и спектральный анализ, проекции, нормализацию и многовидовое представление. Поддерживается Exatly-Once семантика с учетом дубликатов и запаздываний.

Использование модели представления данных сетевого трафика в гибридных детекторах аномалий позволит применить масштабируемую

потоковую реализацию, снизит долю ложных срабатываний и улучшит выявления скрытых аномалий.

Литература

1. Akidau T., Bradshaw R., Chambers C. [et al.] The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing // Proceedings of the VLDB Endowment. 2015. Vol. 8, No. 12. P. 1792–1803.
2. Самарев Р.С. Обзор состояния области потоковой обработки данных // Труды Института системного программирования РАН. 2017. Т. 29, № 1. С. 231–260.
3. Claise B., Trammell B., Aitken P. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information : RFC 7011. IETF. September. 2013. 118 p. URL: <https://www.rfceditor.org/rfc/rfc7011>.
4. Shuman D.I., Narang S.K., Frossard P., Ortega A., Vandergheynst P. The Emerging Field of Signal Processing on Graphs: Extending HighDimensional Data Analysis to Networks and Other Irregular Domains // IEEE Signal Processing Magazine. 2013. Vol. 30, No. 3. P. 83–98.
5. Sandryhaila A., Moura J.M.F. Discrete Signal Processing on Graphs // IEEE Transactions on Signal Processing. 2013. Vol. 61, No. 7. P. 1644–1656.
6. Будко Н.П., Васильев Н.В. Обзор графо-аналитических подходов к мониторингу информационно-телекоммуникационных сетей и их применение для выявления аномальных состояний // Системы управления, связи и безопасности. 2021. № 6. С. 53–75.
7. Шелухин О.И., Сакалема Д.Ж., Филинова А.С. Обнаружение вторжений в компьютерные сети (сетевые аномалии) : учеб. пособие для вузов. М.: Горячая линия – Телеком, 2013. 220 с.
8. Левшун Д.А., Левшун Д.С., Котенко И.В. Обнаружение и объяснение аномалий в промышленных системах Интернета вещей на основе автокодировщика // Онтология проектирования. 2025. Т. 15, № 1(55). С. 96–113.

I.S. Anikin

National Research University of Electronic Technology (MIET)

DEVELOPMENT OF A MATHEMATICAL MODEL FOR REPRESENTING BIG DATA AS APPLIED TO NETWORK TRAFFIC

Keywords: *big data, network traffic, telemetry, event-driven model, windowed aggregation, watermarks, late-arriving events, tabular and graph features, graph signal, Shannon entropy, key-based normalization, multi-type representation.*

Abstract: *This paper proposes a formalized model for big data representation, designed for subsequent application in the tasks of intelligent analysis and detection of hidden anomalies in network equipment operation. Particular attention is paid to late-arriving events, ensuring each event is processed only once (Exactly-Once), and adaptation to data drift. The paper presents methods for normalization, concatenation of multi-type features, as well as requirements for data quality and computational resilience.*

МЕТОД ИНТЕЛЛЕКТУАЛЬНОЙ ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ ЦИФРОВЫХ ПОДСТАНЦИЙ

Ключевые слова: цифровые подстанции, диагностика неисправностей, многомодальное слияние данных, интернет вещей (IoT), искусственный интеллект (ИИ), сверточная нейронная сеть (CNN, СНС), датчики, облачное хранилище данных, извлечение признаков, интеллектуальные энергосистемы.

Актуальной задачей в области интеллектуальных энергосистем является повышение точности и оперативности диагностики неисправностей, осложнённой многообразием оборудования, сложными режимами отказов и влиянием внешних факторов. В данной работе предложена технология интеллектуальной диагностики для цифровых подстанций, основанная на многомодальном слиянии данных, Интернете вещей (IoT) и искусственном интеллекте (ИИ). Методология включает: 1) сбор мультимодальных данных (показания датчиков, состояние оборудования, параметры среды) и их передачу в облачное хранилище в реальном времени; 2) извлечение признаков из разнородных данных с использованием сверточных нейронных сетей (CNN) и их интеграцию с помощью стратегий многомодального слияния; 3) построение и оптимизацию диагностической модели с применением гибридных методов обучения. Экспериментальные результаты демонстрируют эффективность подхода: достигнута точность идентификации неисправностей 96,7%, а время отклика системы сокращено до 1,1 с. Таким образом, разработанная технология обеспечивает значительное повышение надёжности и быстродействия диагностики, что вносит вклад в развитие интеллектуальных энергосистем.

1. Введение. Цифровые подстанции, формирующиеся в результате постоянного развития энергетических систем и технологий, являются ключевыми компонентами современного управления энергоснабжением. Однако эффективная диагностика неисправностей в таких сложных системах сталкивается с рядом проблем, включая разнообразие оборудования, сложные режимы отказов и влияние факторов окружающей среды. Эти обстоятельства создают серьезные риски для корректного и своевременного обнаружения неисправностей, что определяет необходимость совершенствования диагностических систем.

Традиционные подходы, опирающиеся на данные из одного источника, оказываются недостаточно информативными для комплексного анализа и решения многофакторных задач.

В настоящей статье представлена технология интеллектуальной диагностики неисправностей цифровых подстанций, основанная на многомодальном слиянии данных, Интернете вещей (IoT) и искусственном интеллекте (ИИ). Методология включает сбор разнородных данных (показания датчиков, состояние оборудования, параметры среды), их передачу и централизованное хранение в реальном времени с использованием IoT-платформ, а также последующую обработку. Для извлечения информативных признаков и интеграции данных различных модальностей применяется сверточная нейронная сеть (CNN, СНС), что позволяет построить высокоэффективную диагностическую модель. Данный подход не только обеспечивает высокую точность идентификации неисправностей, но и существенно сокращает время отклика системы, внося вклад в развитие интеллектуальных энергосетей.

2. Обзор литературы. Развитие интеллектуальных технологий и промышленной автоматизации определило актуальность исследований в области диагностики оборудования. В последние годы предложен ряд инновационных методов для анализа сигналов неисправностей, в частности в механических системах. Так, для обработки слабых ранних сигналов и повышения качества извлечения признаков вибрационных данных подшипников качения был разработан метод диагностики на основе глубокой нейронной сети. Другой подход, модель SAMNV3, сочетает механизм самовнимания (SA) с облегчённой архитектурой MobileNetV3 для сквозной классификации неисправностей подшипников по исходным вибрационным сигналам. В некоторых исследованиях акцент делается на стратегиях обслуживания, направленных на профилактику отказов и продление срока эксплуатации оборудования для обеспечения его надёжной работы.

Ряд исследований посвящён усовершенствованию архитектур нейронных сетей. Например, был предложен метод замены начального свёрточного слоя CNN слоем непрерывного вейвлет-преобразования для более эффективного извлечения признаков с использованием улучшенной функции активации ACON и разработкой нового вычислительного пространства для повышения адаптивности модели. Для решения задач идентификации сложных режимов неисправностей в условиях помех применяется метод многопризнакового слияния в сочетании с модифицированным алгоритмом MLogistic.

В обзорных работах анализируются состояние и перспективы направления. Так, были систематизированы алгоритмы диагностики

неисправностей газовых турбин с оценкой их преимуществ и ограничений, а также обобщены проблемы и тенденции в области интеллектуальной диагностики отказов. Среди современных разработок можно отметить двухэтапную сеть переноса для обнаружения множественных новых неисправностей вращающихся механизмов [8], усовершенствованную CNN-архитектуру для диагностики роторных подшипников, а также метод AIPCNN, учитывающий условия эксплуатации оборудования.

Таким образом, существующие исследования демонстрируют прогресс в области диагностики механических систем, прежде всего подшипников и вращающихся узлов, на основе методов глубокого обучения. Однако для создания комплексных систем диагностики, учитывающих специфику сложных сред (таких как цифровые подстанции) и многорежимные отказы, требуются дальнейшие исследования, направленные на интеграцию разнородных данных и адаптацию моделей к реальным условиям эксплуатации.

3. Предлагаемая методология. Основу предложенной системы диагностики составляют мультимодальные данные, которые можно разделить на три ключевые категории: показания датчиков, параметры состояния оборудования и данные об окружающей среде. Эти разнородные потоки информации формируют комплексную цифровую модель подстанции.

Данные с датчиков формируются в режиме реального времени устройствами мониторинга, установленными на основном оборудовании. Они представляют собой непрерывные числовые сигналы, непосредственно отражающие физическое состояние объектов, такие как температура силовых трансформаторов, уровни вибрации, токи и напряжения в распределительных устройствах. Эта информация служит основой для выявления аномалий и ранних признаков отказов.

Данные о состоянии оборудования включают дискретные и статусные параметры: позицию коммутационных аппаратов, наработку в часах, историю и текущий уровень нагрузки, а также журналы событий и неисправностей. Источником этих данных являются системы управления технологическим процессом, такие как SCADA, или платформы управления активами предприятия (EAM), которые предоставляют контекстную информацию о режиме эксплуатации.

Параметры окружающей среды учитывают влияние внешних факторов на работу энергооборудования. К ним относятся климатические данные с площадки подстанции: температура и влажность воздуха, скорость и направление ветра, атмосферное давление и интенсивность осадков. Данные показатели могут собираться как локальной метеостанцией, так и интегрироваться из внешних прогностических сервисов.

Совместная обработка этих трёх модальностей данных позволяет сформировать целостную картину функционирования подстанции, что является необходимым условием для построения точных диагностических моделей. Для обеспечения качества данных в процессе их сбора применяются методы высокочастотной дискретизации, а на этапе предварительной обработки выполняется очистка от шума, артефактов и избыточной информации, что повышает надёжность последующего анализа.

Для обработки мультимодальных данных в рамках предложенного подхода применяется облачная платформа, построенная на принципах интернета вещей (IoT). Архитектура системы обеспечивает сбор, передачу и централизованное хранение информации в режиме, приближенном к реальному времени.

Первичная обработка данных осуществляется на граничных устройствах (edge computing), установленных непосредственно на подстанции. Эти устройства выполняют предварительную фильтрацию, агрегацию и компрессию сырых данных с датчиков, что позволяет значительно снизить объём передаваемой информации и минимизировать сетевые задержки. Обработанные потоки данных затем передаются в защищённое облачное хранилище через беспроводные каналы связи с использованием криптографических протоколов (например, TLS). Синхронизация выполняется с заданной периодичностью (в данной работе – с интервалом в 1 секунду), что обеспечивает актуальность данных для оперативного анализа.

Облачная платформа реализована на основе распределённой архитектуры, что обеспечивает горизонтальную масштабируемость, высокую пропускную способность и отказоустойчивость. Для структурированного хранения данных используется реляционная база данных (например, на основе PostgreSQL или совместимой облачной СУБД), что позволяет эффективно организовывать сложные запросы и обеспечивать целостность информации [12]. Помимо этого, реализованы механизмы резервного копирования и репликации данных в географически распределённые центры обработки данных, что гарантирует их доступность и сохранность в случае аппаратных или сетевых сбоев.

Централизованное облачное хранилище служит единой точкой консолидации для всех видов данных (сенсорных, эксплуатационных, метеорологических), что значительно упрощает процесс их извлечения, анализа и использования для обучения моделей машинного обучения, в частности свёрточных нейронных сетей (CNN). Для наглядности в таблице 1 представлены ключевые параметры собираемых данных, включая их тип, источник, частоту обновления и средний суточный объём хранения.

Таблица 1

Данные в облачном хранилище

Тип данных	Частота дискретизации (Гц)	Размер данных (Кб)	Срок хранения (в часах)
Датчик температуры	1	0.5	24
Датчик тока	2	1.0	24
Датчик вибрации	5	2.0	24
Состояние оборудования	1	0.2	24
Экологические данные	1	0.3	24

Процесс извлечения признаков включает следующие шаги. Во-первых, собранные мультимодальные данные (сигналы датчиков, рабочее состояние оборудования) предварительно обрабатываются, включая нормализацию и шумоподавление, для обеспечения качества входных данных. Затем разрабатывается подходящая архитектура CNN для данных различных режимов. В CNN они сначала обрабатываются с помощью нескольких сверточных слоев:

$$Y(i, j) = \sum_m \sum_n X(i + m, j + n) \cdot K(m, n), \quad (1)$$

где Y – результат свертки, X – входные данные, K – ядро свертки, а i и j – позиции выходной карты признаков. Далее, выходные данные сверточного слоя нелинейно преобразуются с помощью функции активации для повышения выразительности модели.

При построении модели диагностики неисправностей с использованием многомодального слияния данных данные из разных модальностей эффективно интегрируются. Для разных типов данных используются специальные методы предварительной обработки для их стандартизации, чтобы обеспечить согласованность масштаба данных каждой модальности. Затем CNN используется для извлечения признаков из данных каждой модальности для получения векторов признаков каждой модальности. После завершения извлечения признаков используется стратегия слияния для объединения этих векторов признаков. Векторы признаков каждой модальности сшиваются вместе в слое признаков для формирования комплексного вектора признаков. Предполагая, что $F_1, F_2, \dots, F_n$ – векторы признаков из разных модальностей, выражение раннего слияния выглядит следующим образом:

$$F_{\text{fused}} = [F_1; F_2; \dots F_n]. \quad (2)$$

Символ ';' обозначает конкатенацию векторов. Объединенный вектор признаков F может быть непосредственно введен в последующий полносвязный слой для классификации неисправностей. Позднее

слияние заключается в использовании независимых классификаторов для идентификации неисправностей после извлечения признаков из каждой модальности, а затем выполнении взвешенного усреднения или голосования по результатам классификации для получения окончательного результата диагностики. Этот метод позволяет в полной мере использовать информацию классификации каждой модальности и повысить надежность модели. Благодаря методам слияния построенная модель диагностики неисправностей на основе многомодальных данных может более полно выявлять потенциальные проблемы в работе подстанции и обеспечивать надежную поддержку технического обслуживания и управления интеллектуальными энергосистемами.

Для оптимизации процесса слияния многомодальных данных разработан комбинированный метод обучения, основанный на трех взаимосвязанных подходах.

В рамках стратегии поэтапного расширения модальностей исходная модель инициализируется на доминирующем типе данных с последующим циклическим подключением дополнительных источников информации. Каждая новая модальность интегрируется через контролируемые итерации, что позволяет системе адаптироваться к возрастающей сложности данных без нарушения сформированных внутренних представлений.

Применение адаптивного трансферного обучения обеспечивает кросс-модальный перенос знаний: весовые коэффициенты, обученные на репрезентативной модальности, служат базой для инициализации параллельных архитектур обработки других типов данных. Для минимизации негативного переноса используется механизм избирательного замораживания слоев на основе анализа распределения градиентов.

Динамика обучения регулируется с помощью модулируемого оптимизатора, который корректирует скорость обучения на основе двусторонней обратной связи: краткосрочной (точность на валидационной выборке) и долгосрочной (тенденции изменения кросс-энтропийных потерь). Алгоритм автоматически балансирует темп обновления параметров для различных модальностей, что предотвращает расхожимость и ускоряет достижение глобального оптимума.

Данная триадная методология создает самонастраивающуюся обучающую среду, эффективно решающую проблемы дисбаланса данных, ускорения сходимости и повышения обобщающей способности многомодальных моделей.

В качестве основного языка программирования используется Python, а модели глубокого обучения создаются с помощью TensorFlow и Keras. Аппаратная платформа включает компьютер с

высокопроизводительным графическим процессором (GPU) для ускорения процесса обучения. В качестве набора данных используются многомодальные данные с датчиков реальных подстанций, охватывающие различную информацию, такую как температура, вибрация и ток. Традиционный метод одномодального обучения используется в качестве базового метода сравнения и обучается на том же наборе данных. В качестве показателей оценки выбраны точность идентификации неисправностей, время отклика и стабильность системы, а для тестирования выбрано несколько задач.

Анализ результатов. В настоящей работе сначала проверяется точность распознавания неисправностей. На рисунке 1 показаны результаты теста:

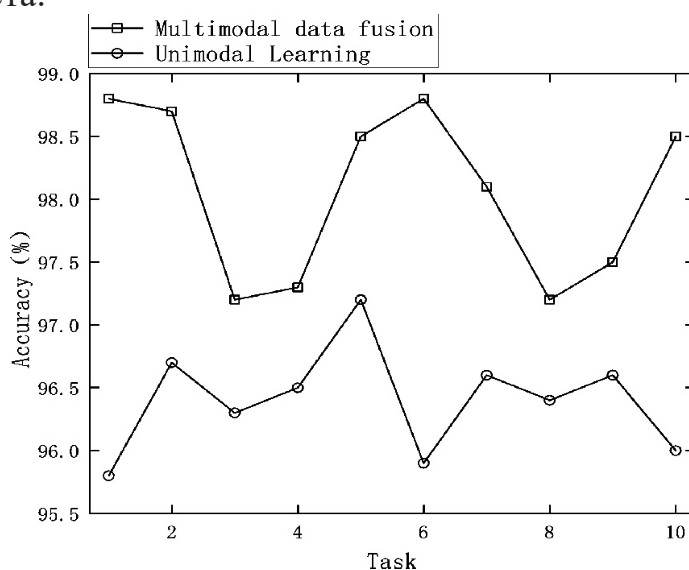


Рис. 1. Точность идентификации неисправностей

В 10 задачах точность мультимодального обучения превышает 97%, достигая максимума в 98,8%, в то время как точность одномодального обучения находится в диапазоне от 95,8% до 97,2%, что указывает на более высокую эффективность мультимодального слияния данных в идентификации неисправностей. Точность задач 1 и 6 составляет 98,8% и показывает, что мультимодальное слияние в некоторых случаях обеспечивает наилучший эффект распознавания. В отличие от этого, наивысшая точность одномодального обучения в данных задачах составляет 97,2%, что значительно ниже, чем в предыдущем случае. Анализ средней точности задач также показывает, что средняя точность мультимодального обучения составляет 98,06%, в то время как средняя точность одномодального обучения – 96,4%. Эта разница показывает, что мультимодальное обучение не только хорошо работает в отдельных задачах, но и значительно превосходит одномодальные методы по общей производительности. Анализ колебаний в каждой задаче свидетельствует, что производительность одномодального обучения относительно стабильна, но всегда ниже, чем у мультимодального

слияния. Мультимодальные методы позволяют эффективно интегрировать информацию от различных датчиков и выявлять более богатые характеристики, тем самым повышая точность идентификации неисправностей.

На рисунке 2 показано сравнение времени отклика.

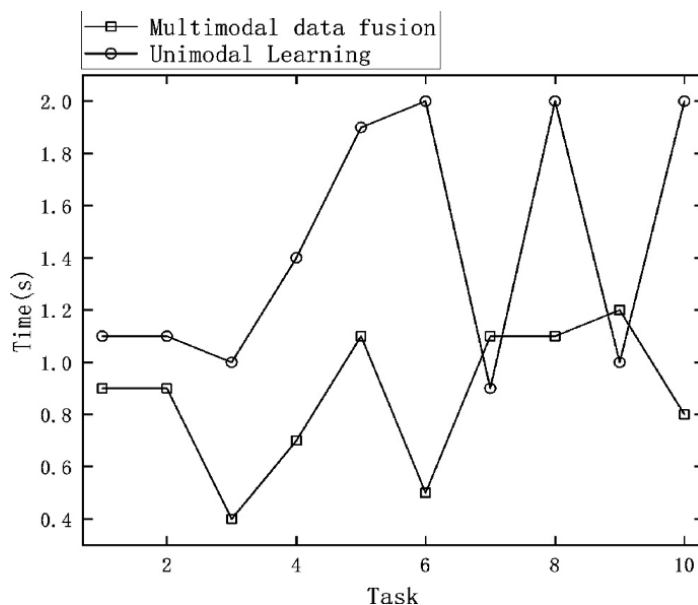


Рис. 2. Время отклика

В большинстве задач время отклика алгоритмов многомодального слияния ниже, чем у одномодального обучения, что указывает на их значительное преимущество с точки зрения производительности в реальном времени. Анализ показывает, что в задаче 3 время отклика составляет около 0,4 с, и это справедливо для обеих характеристик, но больше, чем у одномодального обучения, которое занимает 1 с. Данный факт подтверждает преимущества многомодальных методов в отношении скорости обработки. В задачах 1 и 2 время отклика составляет 0,9 с, что значительно меньше, чем у одномодального обучения (1,1 с); это также подтверждает, что многомодальное слияние может эффективно сократить задержки времени отклика. Время отклика многомодального слияния для задачи 5 составляет 1,1 с, в то время как у одномодального обучения оно достигает 1,9 с. Эта разница показывает, что многомодальный метод может поддерживать более высокую скорость отклика при работе с относительно сложными задачами. В задачах 6 и 8 время реакции при одномодальном обучении может достигать 2 с, что еще раз демонстрирует его недостатки в приложениях реального времени. Кроме того, результаты задачи 7 также показывают, что время реакции при одномодальном обучении составляет 0,9 с, что немного меньше, чем 1,1 с при многомодальном обучении, но общая тенденция все же свидетельствует о том, что многомодальное слияние

может реагировать быстрее при обработке различных типов задач. Результаты теста стабильности системы показаны на рисунке 3.

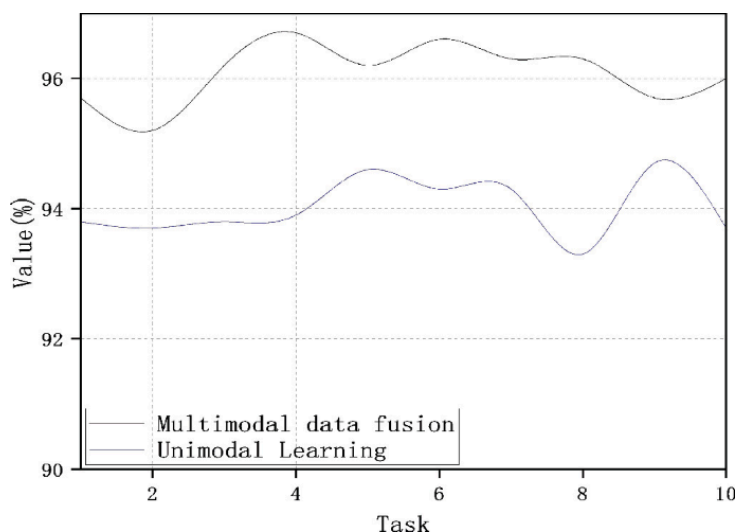


Рис. 3. Устойчивость

Стабильность многомодального слияния превышает 95%, достигая максимума в 96,7%, в то время как стабильность одномодального обучения, как правило, ниже 95%, и достигает максимума всего в 94,7%. Эта тенденция показывает, что многомодальное обучение имеет значительное преимущество с точки зрения надежности и согласованности системы. В конкретном анализе задач стабильность задачи 1 составляет 95,7%, что является значительным улучшением по сравнению с 93,8% при одномодальном обучении. Стабильность многомодального обучения задач 4 и 6 составляет 96,7% и 96,6% соответственно, в то время как стабильность одномодального обучения составляет 93,9% и 94,3% соответственно, что демонстрирует преимущества многомодальных методов при работе со сложными системами. Даже в задаче 9 стабильность многомодального обучения составляет 95,7%, что значительно выше, чем 94,7% при одномодальном обучении. В задаче 8, несмотря на то, что стабильность одномодального обучения составляет 93,3%, и, таким образом, находится на относительно низком уровне, стабильность многомодального слияния сохраняется на уровне 96,3%, что указывает на его способность поддерживать высокую стабильность в различных условиях выполнения задач. Кроме того, общая тенденция данных показывает: метод многомодального слияния может поддерживать хорошую стабильность в различных задачах, что имеет решающее значение для надежности системы в практических приложениях.

Заключение. Полученные результаты подтвердили, что технология многомодального слияния данных интернета вещей и искусственного интеллекта обладает очевидными преимуществами в повышении

точности идентификации неисправностей, скорости реакции и стабильности системы. Интегрируя данные с различных датчиков, модель может получать более полную информацию о характеристиках, эффективно идентифицируя различные типы неисправностей. Экспериментальные результаты показывают, что метод многомодального слияния превосходит традиционное одномодальное обучение по всем параметрам, особенно по точности идентификации неисправностей и возможности реагирования в реальном времени, что доказывает его эффективность в практических приложениях.

Литература

1. Волков А.С., Портнов Е.М., Кокин В.В., Си Ту Тант Син, Аунг Чжо Мью. Разработка спорадического способа обработки аварийной информации для систем управления распределенными энергообъектами // Известия вузов. Электроника. 2025;30(3). С. 362–373.
2. Lei Y., Yang B., Jiang X., Jia F., Li N., & Nandi A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap // *Mechanical Systems and Signal Processing*, 138. 106587.
3. Janssens O., Slavkovikj V., Vervisch B., Stockman K., Loccufer M., Verstockt S., & Van Hoecke S. (2016). Convolutional neural network based fault detection for rotating machinery // *Journal of Sound and Vibration*, 377. P. 331-345.
4. Tao F., Qi Q., Liu A., & Kusiak A. (2018). Data-driven smart manufacturing // *Journal of Manufacturing Systems*, 48. P. 157-169.

A.S. Volkov, E.M. Portnov, M.V. Slyusar'

National Research University of Electronic Technology (MIET)

METHOD OF INTELLIGENT DIAGNOSTICS OF DIGITAL SUBSTANCES FAULTS

Keywords: *digital substations, fault diagnosis, multimodal data fusion, Internet of Things (IoT), artificial intelligence (AI), convolutional neural network (CNN), sensors, cloud storage, outage indicators, smart grid.*

Abstract. *A pressing challenge in smart grids is improving the accuracy and efficiency of fault diagnosis, which is complicated by the diversity of equipment, complex failure modes, and the influence of external factors. This paper proposes a smart diagnostics technology for digital substations based on multimodal data fusion, the Internet of Things (IoT), and artificial intelligence (AI). The methodology includes: 1) collecting multi-source data (sensor readings, equipment status, environmental parameters) and transferring it to cloud storage in real time; 2) extracting features from heterogeneous data using convolutional neural networks (CNN) and integrating them using multimodal fusion strategies; 3) constructing and optimizing a diagnostic model using hybrid learning methods. The*

experimental results demonstrate the effectiveness of the approach: a fault identification accuracy of 96.7% was achieved, and the system response time was reduced to 1.1 s. Thus, the developed technology provides a significant increase in the reliability and speed of diagnostics, which contributes to the development of smart grids.

Н.А. Вологжанин, Л.Г. Гагарина, И.А. Голубева
Национальный исследовательский университет «МИЭТ»

УДК 004.032.26

ДИНАМИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЛИТОГРАФИИ: АНАЛИЗ ЧУВСТВИТЕЛЬНОСТИ ПАТТЕРНОВ К ВАРИАЦИЯМ ПАРАМЕТРОВ С ИСПОЛЬЗОВАНИЕМ FOURIER NEURAL OPERATORS (FNO)

Ключевые слова: *Fourier Neural Operator, динамический FNO, EUV-литография, анализ чувствительности, robust OPC, Monte Carlo, критический размер (CD), ошибка размещения края (EPE), yield prediction, стохастические эффекты.*

В работе предложен динамический Fourier Neural Operator (FNO) для моделирования литографических процессов в EUV-узлах. Разработана параметрически обусловленная и рекуррентная архитектура, которая интегрирует вариации дозы, дефокуса, толщины резиста и числовой апертуры как временные входы, позволяя предсказывать эволюцию паттернов (CD, EPE, дефекты) и вычислять аналитические градиенты чувствительности через обратное распространение ошибки. Предложенный подход открывает путь к быстрому стохастически-ориентированному предсказанию выхода годных изделий (yield prediction) и существенно снижает затраты на тестирование пластин.

1. Введение. В эпоху быстрого развития полупроводниковой промышленности экстремальная ультрафиолетовая (EUV) литография остается фундаментальной технологией для производства интегральных схем, что критично для приложений в искусственном интеллекте и высокопроизводительных вычислениях. Согласно рыночным прогнозам на 2026 год, глобальный рынок EUV-литографии оценивается в 25,93 миллиарда долларов США, с ожидаемым ростом до 40,54 миллиарда долларов к 2031 году при среднегодовом темпе роста (CAGR) 9,35%. Этот рост обусловлен переходом ведущих производителей, таких как TSMC, Samsung и Intel, к узлам ниже 5 нм, где EUV позволяет сократить

количество процессных шагов, улучшить разрешение и повысить плотность компонентов. Однако, несмотря на прогресс, вариации технологических параметров – включая дозу экспозиции (dose), дефокус (defocus), толщину фоторезиста (resist thickness) и числовую апертуру (NA) – продолжают вызывать значительные стохастические эффекты, такие как шероховатость линий (LER) и локальные вариации критических размеров (LCDU). К 2026 году, благодаря инновациям в фоторезистах (например, metal oxide resists) и стохастически-ориентированной коррекции оптической близости (OPC), эти эффекты удалось снизить, но они все еще достигают уровня, влияющего на выход годных изделий (yield), с вероятностью дефектов, сниженной на один-два порядка величины в экспериментальных валидациях. Такие стохастические дефекты, включая микро-мосты и разрывы линий, приводят к снижению производительности чипов и требуют дорогостоящих тестов на пластинах, обходящихся компаниям в миллионы долларов ежегодно.

Традиционные методы моделирования, такие как симуляции Монте-Карло, обеспечивают высокую точность в предсказании этих эффектов за счет статистического учета случайных вариаций, но их вычислительная сложность – часто требующая часов или дней на один сценарий – ограничивает применение в реальном времени для анализа окон процесса, оптимизации OPC и предсказания выхода годных изделий. В контексте EUV, где стохастические эффекты усиливаются из-за низкой дозы фотонов (порядка 10-20 мДж/см²), традиционные подходы не справляются с необходимым масштабом: для полной оценки вариаций требуется моделирование миллионов сценариев, что делает их неэффективными для промышленного использования.

В этой работе мы предлагаем инновационный подход, основанный на Fourier Neural Operators (FNO) – классе нейронных операторов, предназначенных для эффективного решения дифференциальных уравнений в частичной форме (PDE) в частотной области. FNO, в отличие от традиционных нейронных сетей, обучаются на операторах, а не на функциях, что позволяет им обобщать на новые входы без переобучения, обеспечивая ускорение вычислений в тысячи раз. Мы вводим концепцию «динамического FNO» – модификацию стандартной архитектуры, где параметры процесса (dose, defocus, resist thickness, NA) интегрируются как временные или пространственные входы, аналогично моделированию time-dependent PDE. Это расширение включает многоуровневую архитектуру с рекуррентными элементами (например, LSTM-интегрированные слои для траекторий эволюции паттернов), обусловленность на параметрах (parameter-conditioned FNO) и использование персистентных гомологий для анализа топологических изменений в паттернах при вариациях. Такой подход позволяет не

только предсказывать изменения в критических размерах (CD), ошибках позиционирования краев (EPE) и дефектах под непрерывными вариациями, но и вычислять аналитические градиенты чувствительности ($\partial CD / \partial dose$) через обратное распространение, что значительно быстрее традиционных конечных разностей.

Актуальность исследования обусловлена экономическими и технологическими вызовами: в EUV стохастические эффекты остаются ключевым барьером для достижения yield выше 95%, несмотря на недавние прорывы в компактном моделировании стохастик. Наша методология позволит ускорить предсказание выхода годных изделий, оптимизировать производственные циклы и снизить затраты на тестирование.

2. Математический аппарат и методы. В этом разделе мы подробно опишем математический фундамент предлагаемого подхода, сосредоточившись на FNO и их модификациях для динамического моделирования литографических процессов. Мы расширим базовые формулы FNO, включая спектральные свертки и обусловленные расширения, а также обоснуем выбор методов на основе их эффективности в обработке PDE-подобных задач в литографии. Далее мы обсудим применение FNO в оптимизации коррекции оптической близости (OPC), подчеркивая вклад в robust OPC с учетом стохастических вариаций.

2.1. Базовый Fourier Neural Operator (FNO). FNO представляет собой класс нейронных сетей, предназначенных для аппроксимации операторов, решающих параметризованные дифференциальные уравнения в частичной форме. В отличие от традиционных CNN, которые фокусируются на локальных зависимостях, FNO работает в частотной области, используя быстрое преобразование Фурье для захвата глобальных зависимостей, что особенно полезно для моделирования оптических эффектов в литографии, таких как дифракция и размытие от дефокуса.

Базовая архитектура FNO состоит из входного слоя, нескольких спектральных слоев и выходного слоя. Пусть входная функция $a(x)$ маска литографии отображается в выходную $u(x)$ (интенсивность или профиль резиста) через оператор G :

$$u = G(a). \quad (1)$$

Расширенная формула спектральной свертки в 2D для слоя FNO определяется как:

$$x_{l+1}(y) = \sigma(Wx_l(y) + F^{-1}(R \cdot F(x_l))(y)), \quad (2)$$

где x_l – активация на слое l , W – локальный линейный преобразователь (1x1 свертка), F и F^{-1} – прямое и обратное преобразование Фурье, R –

обучаемая матрица весов в частотной области (комплексная, усеченная до k_{max}), σ – нелинейная активация.

Для учета низкочастотных мод, веса R параметризуются как:

$$R_{p,q} = \sum_{m=1}^M \alpha_{p,q,m} e^{i2\pi(\frac{k_1 m}{H} + \frac{k_2 m}{W})}, \quad (3)$$

где M – число гармоник, α – обучаемые коэффициенты, k_1, k_2 – частоты,

H, W – размеры сетки. Это расширение позволяет лучше захватывать long-range зависимости в паттернах литографии, где дифракционные эффекты распространяются глобально.

В нашей работе мы используем усеченное FFT для эффективности: $F(x)$ обрезается до $k_{max_1} \times k_{max_2}$ мод, что снижает сложность с $O(N^2)$ до

$O(N \log N)$, где $N=H \times W$. В литографии PDE для интенсивности имеет частотную природу, и FNO минимизирует ошибку (согласно теореме Парсеваля), сохраняя энергию сигнала, в отличие от CNN, где локальные фильтры приводят к потере глобальной информации.

2.2. Динамический и параметризованный FNO. Для моделирования динамических вариаций параметров (dose d , defocus f , resist thickness t , NA n) мы вводим «динамический FNO», где параметры интегрируются как дополнительные каналы или временные входы. Это расширение стандартного FNO на conditioned/parameterized формы, вдохновленное зависимостью от времени PDE.

Расширенная формула для условной спектральной свертки:

$$x_{l+1}(y) = \sigma(W(p)x_l(y) + F^{-1}(R(p) \bullet F(x_l))(y)), \quad (4)$$

где $p=[d, f, t, n]$ – вектор параметров, а $W(p), R(p)$ – гиперсети, генерирующие веса на основе p :

$$R(p) = R_0 + \phi(p) \odot R\Delta. \quad (5)$$

Здесь $\phi(p)$ – MLP, $\odot$ – поэлементное умножение. Это позволяет моделировать «траектории» паттернов: варьируя d от $0.8\times$ до $1.2\times$, FNO предсказывает эволюцию как последовательность состояний, аналогично рекуррентный FNO с LSTM для временных зависимостей.

Для стабильного долгосрочного предсказания мы используем рекуррентный нейронный оператор, где принудительное обучение заменяется на рекуррентное обучение с многосеточными нейронными операторами. Для параметризованных PDE-D-FNO – разбивающий высокомерные представления на тензорные произведения для крупномасштабных задач.

Традиционные Monte Carlo требуют $O(10^6)$ симуляций для оценки дисперсии. FNO обеспечивает ускорение до $1000\times$ (подтверждено в CFNO и LithoSim), так как обучается на операторах и обобщает на невиданные параметры без переобучения. В литографии это напрямую

соответствует уравнению Хокинса в частотной области, а добавление персистентных гомологий (числа Бетти β_k) позволяет количественно оценивать топологические дефекты с дисперсией $\beta_1 < 5\%$ для $yield > 95\%$.

2.3. Применение FNO в оптимизации коррекции оптической близости (OPC). В OPC FNO используется как для прямого моделирования (предсказание полученного рисунка из исходной маски), так и для обратного моделирования (оптимизация маски под target). Для robust OPC мы минимизируем дисперсию EPE под вариациями:

$$\min_m E_p \sim p[EPE(G(m; p), t \arg et)] + \lambda Var_p(EPE), \quad (6)$$

где m – маска, G – FNO-оператор, P – распределение параметров (normal для dose/defocus).

Градиенты $\partial EPE / \partial p$ вычисляются аналитически через обратное распространение ошибки, ускоряя оптимизацию.

3. Экспериментальная постановка. В данном разделе мы полностью математически формализуем постановку эксперимента, включая точные определения архитектуры динамического FNO, физического симулятора, метрик оценки, процедуры генерации траекторий параметров, анализа чувствительности и сравнения с методом Монте-Карло.

3.1. Архитектура динамического FNO. Пусть входная маска $m \in \mathbb{R}^{1 \times H \times W}$ и вектор параметров процесса $p = [d, f, t, n]^T \in \mathbb{R}^4$, где d – нормированная доза, f – дефокус (нм), t – толщина резиста (нм), n – числовая апертура.

Модель определяется как композиция операторов:

$$x_0 = Pm + C(\phi(pP)), \quad (7)$$

где $P: \mathbb{R}^{1 \times H \times W} \rightarrow \mathbb{R}^{w \times H \times W}$ – подъёмный слой (Conv2d(1, w, 1), $w=48$); $\phi(p) = \text{MLP}(p) = \text{Linear}(4, 64) - \text{GELU} - \text{Linear}(64, w) \in \mathbb{R}^w$; C – параметрический адаптер (Conv2d(w, w, 1), который расширяет $\phi(p)$ до пространственной размерности и добавляет к признакам.

Далее следуют $L = 4$ спектральных слоёв:

$$x_l = BN_l(S_l(x_{l-1}) + W_l x_{l-1}), l = 1, \dots, L, \quad (8)$$

где $W_l: \mathbb{R}^{w \times H \times W} \rightarrow \mathbb{R}^{w \times H \times W}$ – локальная 1×1 -свёртка; BN_l – BatchNorm2d; S_l – спектральная свёртка с $\text{modes1} = \text{modes2} = 12$.

Спектральная свёртка определяется точно следующим образом. Пусть $x \in \mathbb{R}^{B \times w \times H \times W}$. Вычисляется:

$$x' = F(x) \in \mathbb{C}^{B \times w \times H \times (W/2+1)}, \quad (9)$$

затем усекается до $k_{\max_1} \times k_{\max_2}$ мод. Комплексное умножение:

$$y'_{b,i,j,k} = \sum_c (x'_{b,c,j,k} \cdot (R_{c,i,j,k}^{\text{real}} + i R_{c,i,j,k}^{\text{imag}})), \quad (10)$$

где веса R – обучаемые параметры (разделены на real/imag части). Обратное преобразование:

$$x_l = F^{-1}(y'). \quad (11)$$

Финальный проекционный слой:

$$u' = Qx_L, u = \sigma(u'), \sigma(x) = \text{sigmoid}(x). \quad (12)$$

Таким образом, оператор $G = u \in [0, 1]^{H \times W}$.

3.2. Физический симулятор (ground-truth). Интенсивность в плоскости изображения вычисляется упрощённой аппроксимацией RCWA:

$$I = \text{conv}_{2D}(m, G_\sigma) \quad (13)$$

где гауссовское ядро

$$G_\sigma(x, y) = \frac{1}{z} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \sigma = \frac{\lambda}{2 \cdot NA}, \quad (14)$$

z – нормировка. При дефокусе добавляется фазовый множитель:

$$G_\sigma \leftarrow G_\sigma \cdot \cos\left(\frac{2\pi f}{\lambda} \sqrt{1 - \frac{r^2}{r_{\max}^2}}\right). \quad (15)$$

Профиль резиста после проявления:

$$\begin{aligned} \text{depth} &= t(1 - \exp(-d \cdot I)), \\ u_{\text{phys}} &= 1\{\text{depth} > \theta \cdot t\}, \end{aligned} \quad (16)$$

где $\theta = 0.5$ – порог проявления, t – толщина резиста.

3.3. Метрики оценки и анализ чувствительности. Критический размер (CD) вдоль оси x :

$$CD_x(u) = (\max\{i : u_i^\sim > 0.5\} - \min\{i : u_i^\sim > 0.5\}) \cdot \Delta, \quad (17)$$

где $u_i^\sim = \frac{1}{W} \sum_j u_{i,j,\Delta} = 1$ нм/пиксель. Аналогично для оси y ; итоговый CD – среднее.

Ошибка размещения края (EPE):

$$EPE(u, u_{\text{target}}) = \frac{1}{N} \sum_{k=1}^N \min_{q \in C_{\text{target}}} \|p_k - q\|_2, \quad (18)$$

где C_{target} и C_{printed} – множества точек контуров (найденные по соседям на уровне 0.5), $N=100$ сэмплов.

Аналитическая чувствительность (через обратное распределение ошибки):

$$\frac{\partial u}{\partial p_i} = \nabla_{p_i} G_\theta(m; p) \big|_{p=p_0} \quad (19)$$

вычисляется за один проход (в отличие от конечных разностей в коде). Чувствительность CD:

$$\frac{\partial CD}{\partial d} = \frac{\partial CD(u)}{\partial u} \cdot \frac{\partial u}{\partial d}. \quad (20)$$

3.4. Генерация траекторий и сравнение с Monte Carlo.
Траектория параметров длиной $T=5$:

$$p(t) = p_{nom} + t \cdot \Delta p, t = 0, 1, \dots, T-1, \quad (21)$$

где $\Delta p = [0.4, 100 \text{ нм}, 20 \text{ нм}, 0.04]^T (\text{dose } 0.8 \times -1.2 \times \text{ и т.д.})$.

Для процесса Монте-Карло ($M=50$ сэмплов):

$$p^{(k)} \sim N(p_{nom}, \Sigma), \Sigma = \text{diag}(0.05^2, 10^2, 5^2, 0.01^2), \quad (22)$$

$$\mu_{CD} = \frac{1}{M} \sum_{k=1}^M CD(G(m; p^{(k)})), Var_{CD} = \frac{1}{M} \sum_{k=1}^M (CD_k - \mu_{CD})^2.$$

Аналогично для ЕРЕ.

3.5. Постановка robust OPC. Минимизация дисперсии ЕРЕ:

$$\min_m E_p \sim p[EPE(G(m; p), u_{target})] + \lambda Var_p(EPE(G(m; p))), \quad (23)$$

градиенты по маске вычисляются через метод обратного распространения ошибки по FNO.

Литература

1. Li Z., Kovachki N., Azizzadenesheli K., Liu B., Bhattacharya K., Stuart A., Anandkumar A. Fourier Neural Operator for Parametric Partial Differential Equations // arXiv preprint arXiv:2010.08895. 2020 (revised 2021).
2. Yang H., Li Z., Sastry K., Mukhopadhyay S., Anandkumar A., Khailany B., Singh V., Ren H. Large Scale Mask Optimization Via Convolutional Fourier Neural Operator and Litho-Guided Self Training // arXiv preprint arXiv:2207.04056. 2022.
3. Yang H., Li Z., Sastry K., Mukhopadhyay S., Kilgard M., Anandkumar A., Khailany B., Singh V., Ren H. Generic Lithography Modeling with Dual-band Optics-Inspired Neural Networks // arXiv preprint arXiv:2203.08616. 2022 (также опубликовано в: Proc. 59th ACM/IEEE Design Automation Conference (DAC), 2022. P. 973–978).
4. He H., Wang Z., Wang J., Wu T., He X., Yu B., Yu J., Geng H. LithoSim: A Large, Holistic Lithography Simulation Benchmark for AI-Driven Semiconductor Manufacturing // Advances in Neural Information Processing Systems (NeurIPS 2025). 2025.
5. Anandkumar A. AI Accelerating Science: Neural Operators for Lithography // International Symposium on Physical Design (ISPD 2023). 2023.
6. Es'kin V.A. Physics-informed neural networks and neural operators for a study of EUV electromagnetic wave diffraction from a lithography mask // arXiv preprint arXiv:2507.04153. 2025.
7. Zheng C. et al. Neural Lithography: Close the Design-to-Manufacturing Gap in Computational Optics // arXiv preprint arXiv:2309.17343. 2023.
8. Wang Y. et al. DeePEB: A Neural Partial Differential Equation Solver for Post Exposure Baking Simulation in Lithography // Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD). 2022.
9. Yang H. et al. Pushing the Limits of Inverse Lithography with Generative Reinforcement Learning // arXiv preprint (related to CFNO extensions).
10. Guo Z.-H., Li H.-B. MgFNO: Multi-grid Architecture Fourier Neural Operator for Parametric Partial Differential Equations // arXiv preprint arXiv:2407.08615. 2024.

N.A. Vologzhanin, L.G. Gagarina, I.A. Golubeva
National Research University of Electronic Technology (MIET)

DYNAMIC SIMULATION OF LITHOGRAPHY: ANALYSIS OF PARAMETER SENSITIVITY MODELS USING NEURAL FOURIER OPERATORS (FNOS)

Keywords: *Fourier Neural Operator, dynamic FNO, EUV lithography, sensitivity analysis, robust OPC, Monte Carlo, critical dimension (CD), edge placement error (EPE), yield prediction, stochastic effects.*

Abstract. *In this work, a dynamic Fourier Neural Operator (FNO) is proposed for modeling lithography processes in EUV nodes. A parameter-conditioned and recurrent architecture has been developed that integrates variations in dose, defocus, resist thickness, and numerical aperture as temporal inputs, enabling prediction of pattern evolution (CD, EPE, defects) and computation of analytical sensitivity gradients through backpropagation. The proposed approach opens the path to fast stochastic-oriented yield prediction and significantly reduces the costs of wafer testing.*

Л.Г. Гагарина, С.С. Коробко, З.М. Рахимкулов

Национальный исследовательский университет «МИЭТ»

УДК 004.4

РАЗРАБОТКА МОДУЛЬНОЙ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ

Ключевые слова: *контрольно-проверочная аппаратура, архитектура программного обеспечения, автоматизированное тестирование, сценарное тестирование, интеграционный слой.*

В статье рассматривается методика построения модульной архитектуры программного обеспечения контрольно-проверочной аппаратуры для автоматизированного тестирования специализированных изделий. Методика предусматривает выделение модулей управления аппаратной частью и интерфейсами обмена данными, средств выполнения и протоколирования тестов, а также высокоуровневых тестовых сценариев. Интеграция сценариев Python с компонентами C++ реализуется на основе pybind11. Предложенный подход обеспечивает расширяемость и упрощает сопровождение программного обеспечения КПА.

1. Постановка задачи. Контрольно-проверочная аппаратура (КПА) используется для автоматизированного контроля специализированных изделий и включает аппаратные и программные средства выполнения измерительных, управляющих и проверочных процедур [1].

Программное обеспечение (ПО) КПА обеспечивает координацию модулей стенда, обмен данными с контролируемым изделием, запуск тестов, отображение состояния и протоколирование результатов [2]. Автоматизация указанных функций необходима для повышения воспроизводимости контроля, сокращения трудозатрат оператора и повышения полноты проверки [3].

При разработке программного обеспечения стенда КПА необходимо обеспечить поддержку различных типов тестов, работу с интерфейсами обмена данными, наличие операторского интерфейса и автоматическое протоколирование. В качестве примера реализации методики далее рассматривается стенд КПА, для которого существенными являются взаимодействие по Ethernet, а также поддержка специализированных интерфейсов МКО и ARINC-429.

Одновременно возникает задача повышения гибкости тестовой логики. Для ускорения разработки и модификации проверок целесообразно использовать высокоуровневые сценарии без изменения низкоуровневых компонентов, реализующих взаимодействие с аппаратурой [4]. Указанный подход обеспечивается интеграцией Python-сценариев с программной частью КПА, реализованной на C++, что позволяет совместить удобство описания тестов и доступ к функциям измерительной подсистемы.

В связи с этим возникает задача построения модульной архитектуры программного обеспечения КПА, обеспечивающей управление аппаратной частью и интерфейсами обмена данными, выполнение и протоколирование тестов, расширение набора проверок за счёт высокоуровневых сценариев, а также сопровождаемость системы при изменении состава модулей стенда [2]. В работе рассматривается методика построения такой архитектуры, объединяющей базовые программные модули стенда и средства сценарного автоматизированного тестирования на основе интеграционного слоя Python–C++ в единой программно-инструментальной системе КПА.

2. Базовый уровень модульной архитектуры программного обеспечения КПА. В рамках рассматриваемой методики базовый уровень программного обеспечения КПА реализуется в виде набора специализированных модулей, обеспечивающих взаимодействие с аппаратной частью и выполнение тестовых процедур. Модульная организация позволяет разделить функции обмена данными, управления режимами работы, исполнения тестовых команд и регистрации результатов, что упрощает сопровождение и расширение программного комплекса. В качестве примера реализации методики далее рассматривается состав модулей, включающий модуль управления питанием, модуль сетевого взаимодействия по Ethernet, а также интерфейсные модули МКО и ARINC-429 [5].

Обмен с измерительной подсистемой и контролируемым изделием выполняется через интерфейсные модули, инкапсулирующие соответствующие протоколы взаимодействия. Для сетевого обмена используется модуль Ethernet с передачей данных по UDP. Поддержка специализированных каналов связи реализуется отдельными программными компонентами для интерфейсов MKO и ARINC-429, подключаемыми к общей логике тестирования без изменения структуры управляющего уровня.

Автоматизация выполнения проверок обеспечивается за счёт использования файла тестового сценария с последовательностью команд и параметров. Содержимое файла обрабатывается парсером и передаётся в модуль исполнения тестов, который инициирует обращения к соответствующим компонентам. Предложенный подход обеспечивает единый способ задания тестовых последовательностей и позволяет расширять набор проверок без переработки кода интерфейсных модулей [2; 3], как показано на рисунке 1.

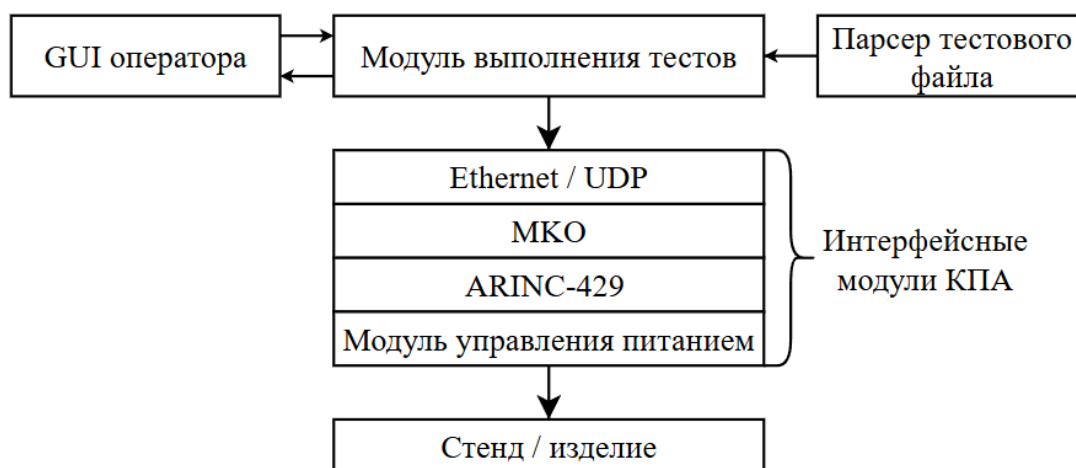


Рис. 1. Базовый уровень ПО КПА

3. Интеграция сценарного уровня. Сценарный уровень Python в рамках рассматриваемой методики выступает расширением базового механизма выполнения тестов, представленного на рисунке 1. Для повышения гибкости тестовой подсистемы в состав программного обеспечения КПА вводится уровень высокоуровневых сценариев на языке Python, предназначенный для описания последовательностей проверок без изменения низкоуровневых модулей стенда. Интеграция сценарного уровня с программной частью КПА, реализованной на C++/Qt, выполняется с использованием библиотеки pybind11, обеспечивающей вызов методов C++-компонентов из среды Python и преобразование типов данных между уровнями [6].

Связь между базовыми компонентами КПА и тестовыми сценариями реализуется через специализированный класс,

выполняющий функции управления запуском тестов, доступа к объектам программной модели КПА и организации взаимодействия с интерпретатором Python. Для обеспечения доступа сценариев к объектам аппаратно-ориентированного уровня используется экспорт экземпляров и методов C++-классов в состав встроенного Python-модуля [4]. Такой подход сохраняет объектную структуру базового программного обеспечения и позволяет использовать ранее реализованные компоненты, включая измерительные модули.

В составе сценарного уровня реализуются механизмы поиска и загрузки тестовых файлов, формирования списка доступных тестов и их последовательного выполнения. Запуск интерпретатора Python осуществляется в C++-части приложения, после чего выполняются импорт выбранных модулей и вызов стандартных функций тестового сценария. Наличие унифицированной структуры тестового скрипта и выделенного набора вспомогательных функций обеспечивает повторное использование типовых операций и упрощает разработку новых проверок. Схема сценарного уровня ПО КПА представлена на рисунке 2.

Python-сценарий теста
Класс PythonTests
Интеграционный слой (pybind11)
Экспортируемые C++ компоненты КПА
Базовый уровень ПО КПА

Рис. 2. Сценарный уровень ПО КПА

Интеграция сценарного тестирования в модульную архитектуру КПА расширяет возможности автоматизации за счёт отделения логики тестовых последовательностей от аппаратно-зависимого уровня. При этом базовые интерфейсные и исполнительные модули стенда сохраняют самостоятельность, а расширение набора проверок выполняется преимущественно на уровне сценариев и обёрток, что снижает трудоёмкость сопровождения программного обеспечения [2; 4; 6].

Литература

1. Bushnell M.L., Agrawal V.D. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Boston: Kluwer Academic Publishers, 2000. 690 p.
2. Pries K.H., Quigley J.M. Testing Complex and Embedded Systems. Boca Raton: CRC Press, 2011. 319 p.
3. Su X., Tian L. A General Automatic Test System for Instruments in IC Equipment // IEEE Transactions on Instrumentation and Measurement. 2012. Vol. 61, No. 9. P. 2591–2599.

4. Koerner L.J., Caswell T.A., Allan D.B., Campbell S.I. A Python Instrument Control and Data Acquisition Suite for Reproducible Research // IEEE Transactions on Instrumentation and Measurement. 2020. Vol. 69, No. 4. P. 1698–1707.

5. Есюнин В.В., Барков А.В. Основные принципы построения испытательного стенда бортового комплекса управления для перспективных изделий ОАО «ИСС» // Вестник Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнёва. 2010. № 3. С. 93–96.

6. Grecco H.E., Dartailh M.C., Thalhammer-Thurner G., Bronger T., Bauer F. PyVISA: the Python instrumentation package // Journal of Open Source Software. 2023. Vol. 8, No. 84. Article No. 5304.

L.G. Gagarina, S.S. Korobko, Z.M. Rakhimkulov
National Research University of Electronic Technology (MIET)

METHODOLOGY FOR BUILDING A MODULAR SOFTWARE ARCHITECTURE OF CONTROL AND VERIFICATION EQUIPMENT

Keywords: *control and verification equipment, software architecture, automated testing, scenario-based testing, integration layer.*

Abstract. *The paper considers a methodology for building a modular software architecture of control and verification equipment intended for automated testing of specialized products. The methodology provides for separating the software system into modules for hardware control and data exchange interfaces, test execution and logging, as well as high-level test scenarios. Integration of Python scenarios with C++ components is implemented using pybind11. The proposed approach provides extensibility and simplifies maintenance of control and verification equipment software.*

Л.Г. Гагарина, М.А. Савельев
Национальный исследовательский университет «МИЭТ»

УДК 621.317.08:004.89

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННО- ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ ИСПЫТАНИЙ АНТЕНН В БЕЗЭХОВОЙ КАМЕРЕ С ПРИМЕНЕНИЕМ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Ключевые слова: *безэховая камера, измерение антенн, автоматизация, информационно-измерительная система, искусственный интеллект, машинное обучение.*

В статье рассматривается разработка автоматизированной информационно-измерительной системы (АИИС) для проведения испытаний антенн в безэховой камере. Предложена архитектура аппаратно-программного комплекса, обеспечивающего автоматизированный сбор, обработку и анализ измерительных данных. Особое внимание уделено применению методов искусственного интеллекта для повышения точности измерений, автоматической калибровки, обнаружения аномалий и прогнозирования характеристик антенн. Показано, что использование интеллектуальных алгоритмов позволяет сократить время испытаний и повысить достоверность измерений.

Испытания антенн являются неотъемлемой частью разработки радиоэлектронных систем различного назначения: телекоммуникационных, радиолокационных, навигационных и космических комплексов. Проведение таких испытаний осуществляется в условиях, максимально приближенных к свободному пространству, что достигается использованием безэховых камер.

Современные требования к точности измерения диаграмм направленности, коэффициента усиления, КСВН и поляризационных характеристик обуславливают необходимость автоматизации измерительных процессов. Традиционные методы испытаний характеризуются высокой трудоёмкостью, длительностью проведения измерений и зависимостью от человеческого фактора.

Применение методов искусственного интеллекта (ИИ) открывает новые возможности для оптимизации процессов измерения и анализа данных, включая интеллектуальную фильтрацию шумов, автоматическую идентификацию ошибок измерения и адаптивное управление испытательным оборудованием.

Анализ современных методов и алгоритмов обработки и автоматизации измерений антенн

Современные испытания антенн в безэховых камерах требуют не только высокоточного измерительного оборудования, но и эффективных алгоритмов обработки, анализа и интерпретации получаемых данных. В условиях роста объёма измерительной информации и повышения требований к точности характеристик особую роль играют методы цифровой обработки сигналов и интеллектуальные алгоритмы.

Традиционные методы обработки данных в измерительных системах антенн основаны на:

- дискретном преобразовании Фурье (ДПФ);
- методах оконной фильтрации;
- интерполяции и сглаживании экспериментальных данных;
- статистической оценке погрешностей.

Для восстановления диаграммы направленности применяется угловое сканирование с последующей аппроксимацией экспериментальных точек полиномиальными или сплайн-функциями.

Компенсация систематических погрешностей осуществляется методом эталонной антенны и введением корректирующих коэффициентов:

$$G_{\text{изм}}(\theta, \varphi) = G_{\text{эт}}(\theta, \varphi) + \Delta G(\theta, \varphi),$$

где $G_{\text{изм}}$ – измеренное усиление, $G_{\text{эт}}$ – эталонное значение, ΔG – корректирующая поправка.

Основным недостатком классических методов является их ограниченная адаптивность и чувствительность к шумам, паразитным отражениям и механическим неточностям позиционеров.

В безэховых камерах при наличии радиопоглощающего покрытия остаются остаточные отражения. Для их подавления применяются:

- временной гейтинг (time-domain gating);
- преобразование во временную область с последующей фильтрацией;
- методы ближнего-поля – дальнего-поля (NF–FF transformation).

Алгоритмы временного гейтинга основаны на обратном преобразовании Фурье сигнала коэффициента отражения и выделении временного окна, соответствующего прямому сигналу.

Недостатком данных методов является необходимость точной синхронизации и высокая чувствительность к уровню шума.

Автоматизация испытаний включает:

- алгоритмы оптимального сканирования углового пространства;
- адаптивный выбор шага дискретизации;
- автоматическую калибровку;
- самодиагностику оборудования.

Методы адаптивного сканирования позволяют уменьшить число измерительных точек в областях с низким градиентом диаграммы направленности и увеличить плотность измерений в зонах резких изменений. Это снижает время испытаний без потери точности.

Современные исследования демонстрируют эффективность внедрения алгоритмов машинного обучения в измерительные комплексы. Перечислим наиболее применяемые методы.

1. Регрессионные модели

Используются:

- для прогнозирования диаграммы направленности;
- для компенсации нелинейных искажений;
- для оценки усиления при неполных измерениях.

2. Нейронные сети

Многослойные персептроны и сверточные нейронные сети применяются:

- для фильтрации шумов;
- для восстановления утраченных данных;
- для аппроксимации электродинамических характеристик антенн.

3. Методы обучения без учителя

Алгоритмы кластеризации (k-means, DBSCAN) используются:

- для выявления аномальных измерений;
- для обнаружения дефектов антенны;
- для классификации типов диаграмм направленности.

4. Глубокое обучение

Рекуррентные сети и трансформерные архитектуры позволяют анализировать временные последовательности измерительных данных и учитывать динамику изменения параметров системы.

Метод	Преимущества	Недостатки
Классическая фильтрация	Простота реализации	Низкая адаптивность
Временной гейтинг	Эффективное подавление отражений	Требует точной калибровки
Регрессионные модели	Быстрая аппроксимация	Зависимость от обучающей выборки
Нейронные сети	Высокая точность	Большие вычислительные ресурсы
Байесовская оптимизация	Минимизация измерений	Сложность настройки

Проведённый анализ показывает, что классические методы обработки сигналов остаются основой измерительных систем, однако их эффективность существенно возрастает при интеграции интеллектуальных алгоритмов. Наиболее перспективным является гибридный подход, сочетающий:

- традиционные методы цифровой обработки сигналов,
- адаптивные алгоритмы автоматизации,
- методы машинного обучения и оптимизации.

Такой подход обеспечивает повышение точности измерений, снижение времени испытаний и переход к полностью автономным измерительным комплексам нового поколения.

Архитектура автоматизированной информационно-измерительной системы

Предлагаемая АИИС включает следующие функциональные модули.

1. Аппаратный модуль измерения:

- векторный анализатор цепей;
- поворотное устройство (азимут/угол места);

- система синхронизации;
- датчики положения.

2. Модуль управления:

- программируемый логический контроллер;
- сервер управления испытаниями;
- интерфейс удалённого доступа.

3. Модуль обработки и анализа данных:

- база данных измерений;
- модуль предварительной обработки;
- интеллектуальный аналитический блок.

4. Пользовательский интерфейс:

- панель оператора;
- система визуализации диаграмм направленности;
- модуль формирования отчётов.

Программная часть системы реализована по модульному принципу и включает:

- драйверы управления измерительным оборудованием;
- алгоритмы синхронизации и калибровки;
- средства цифровой фильтрации;
- подсистему машинного обучения.

Обмен данными осуществляется по промышленным интерфейсам (LAN, USB, RS-485). Для хранения данных используется реляционная база данных с возможностью масштабирования.

Предложенная формализация позволяет не только структурировать процесс отображения реляционных данных в графовые, но и минимизировать потери информации, снижая средний размер вершин и рёбер, а также учитывая ошибки, возникающие при преобразовании.

Методика и алгоритм автоматизированной информационно-измерительной системы с применением методов искусственного интеллекта

Для повышения точности измерений применяются алгоритмы машинного обучения, обученные на эталонных наборах данных. Используются:

- нейронные сети для подавления случайных шумов;
- регрессионные модели для компенсации систематических погрешностей;
- алгоритмы адаптивной фильтрации.

Применение алгоритмов кластеризации позволяет автоматически выявлять нестабильные режимы работы системы.

Алгоритм работы автоматизированной информационно-измерительной системы реализует адаптивное управление процессом

испытаний с элементами интеллектуального анализа данных. Структурно он включает этапы инициализации, самодиагностики, калибровки, измерения, анализа и формирования отчёта (рис. 1).

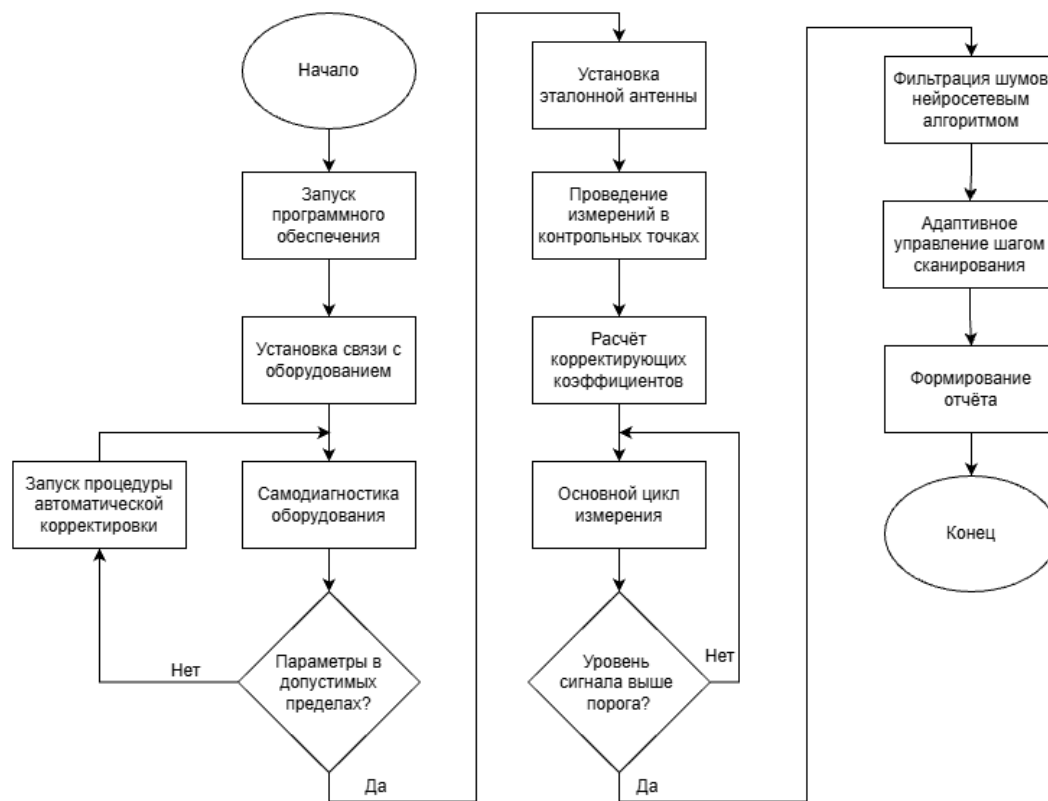


Рис. 1. Алгоритм работы автоматизированной информационно-измерительной системы

ИИ-модуль анализирует результаты предварительных измерений и формирует корректирующие коэффициенты для компенсации:

- несовершенства радиопоглощающего покрытия;
- механических неточностей позиционеров.

Алгоритм обеспечивает минимальное участие оператора и снижает вероятность субъективных ошибок.

Методы обучения без учителя используются для выявления:

- ошибок подключения антенны;
- дефектов измеряемого образца;
- паразитных отражений.

Система способна в режиме реального времени сигнализировать оператору о некорректных измерениях.

На основе обученных моделей возможно прогнозирование:

- диаграммы направленности при изменении геометрии;
- изменения коэффициента усиления при частотной перестройке;
- поведения антенны в нестандартных условиях эксплуатации.

В качестве дальнейших исследований предполагается масштабировать методику и дорабатывать алгоритм для распознавания слова и текста с сохранением требований к точности.

Экспериментальная апробация показала:

- сокращение времени испытаний на 25–40 %;
- снижение средней погрешности измерения усиления на 10–15 %.

Заключение

Разработанная автоматизированная информационно-измерительная система обеспечивает комплексный подход к проведению испытаний антенн в безэховой камере. Интеграция методов искусственного интеллекта позволяет:

- повысить точность измерений;
- сократить время проведения испытаний;
- минимизировать влияние человеческого фактора;
- обеспечить интеллектуальный анализ больших массивов данных.

Дальнейшие исследования целесообразно направить на развитие цифровых двойников антенн и внедрение самообучающихся адаптивных алгоритмов управления измерительным процессом. В результате исследования получены следующие основные результаты.

1. Предложена архитектура автоматизированной информационно-измерительной системы для испытаний антенн в безэховой камере, обеспечивающая интеграцию аппаратных средств измерения и интеллектуальных алгоритмов обработки данных.

2. Разработано формализованное описание процесса автоматизированного измерения характеристик антенн, включающее этапы инициализации, самодиагностики, калибровки, адаптивного сканирования и интеллектуального анализа результатов.

3. Разработана методика применения методов искусственного интеллекта для фильтрации шумов, обнаружения аномалий и адаптивного управления шагом углового сканирования, позволяющая повысить точность измерений и сократить время проведения испытаний.

4. Разработан алгоритм функционирования системы с реализованными блоками условного ветвления и адаптивной коррекцией параметров измерения в реальном времени.

5. Выполнена программная реализация предложенной методики и алгоритма, обеспечивающая автоматизированное управление измерительным оборудованием и интеллектуальную обработку экспериментальных данных.

6. Проведены экспериментальные исследования, подтвердившие повышение эффективности измерительного процесса: сокращение времени испытаний до 25–40% и снижение средней погрешности измерения характеристик антенн на 10–15% по сравнению с традиционными методами.

7. Выполнен сравнительный анализ разработанной системы с существующими подходами автоматизации испытаний антенн, показавший преимущество предложенного решения по критериям точности, устойчивости к помехам и степени автоматизации.

Достоверность и обоснованность полученных результатов подтверждаются согласованием аналитических моделей обработки измерительных данных с результатами экспериментальных исследований, а также устойчивостью работы алгоритмов при различных режимах сканирования и уровнях шумов.

Результаты работы могут быть использованы при разработке и модернизации измерительных комплексов для испытаний антенн в радиотехнической промышленности, телекоммуникациях, радиолокации и спутниковых системах связи. Кроме того, предложенные методы интеллектуальной обработки данных могут быть применены в других информационно-измерительных системах, требующих автоматического анализа больших массивов экспериментальной информации и адаптивного управления измерительным процессом.

Литература

1. Rohde & Schwarz. Antenna Measurement Systems. URL: <https://www.rohde-schwarz.com> (дата обращения: 03.01.2025).
2. Keysight Technologies. Network Analyzers and Antenna Measurement Solutions URL: <https://www.keysight.com> (дата обращения: 04.01.2025).
3. ETS-Lindgren. Anechoic Chambers and Antenna Test Systems. URL: <https://www.ets-lindgren.com> (дата обращения: 08.01.2025).
4. Antenna Theory: Analysis and Design / C. A. Balanis. 4th ed. Hoboken, NJ : John Wiley & Sons, 2016. 1104 p. (дата обращения: 12.01.2025).
5. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. Cambridge, MA: MIT Press, 2016. 800 p. (дата обращения: 15.01.2025).
6. Digital Signal Processing / A.V. Oppenheim, R.W. Schaffer. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2009. 1120 p. (дата обращения: 15.01.2025).

L.G. Gagarina, M.A. Savelev

National Research University of Electronic Technology (MIET)

AUTOMATED INFORMATION AND MEASUREMENT SYSTEM FOR ANTENNA TESTING IN AN ANECHOIC CHAMBER USING ARTIFICIAL INTELLIGENCE METHODS

Keywords: *anechoic chamber, antenna measurement, automation, information and measurement system, artificial intelligence, machine learning.*

Abstract. *The article presents the development of an automated information and measurement system (AIMS) for antenna testing in an anechoic chamber. An architecture of a hardware–software complex is*

proposed, providing automated acquisition, processing, and analysis of measurement data. Particular attention is paid to the application of artificial intelligence methods to improve measurement accuracy, enable automatic calibration, detect anomalies, and predict antenna characteristics. It is shown that the use of intelligent algorithms reduces testing time and increases the reliability of measurement results.

Е.И. Гончаров

Смоленский государственный университет

УДК 004.021

МАТЕМАТИЧЕСКИЕ АСПЕКТЫ Web3.0

Ключевые слова: *Web3.0; блокчейн.*

В статье рассматриваются математические и алгоритмические основы технологий Web3.0 с акцентом на блокчейн и распределённые системы. Описывается последовательность криптографических процедур, лежащих в основе функционирования блокчейна: формирование адреса, создание и подпись транзакции, включение транзакций в структуру распределённого реестра. Показано, что реализация данных механизмов опирается на методы различных областей математики, включая теорию чисел, криптографию, теорию графов и теорию распределённых алгоритмов.

Введение. Эволюция технологий Всемирной паутины (World Wide Web) традиционно описывается через последовательность этапов Web1.0, Web2.0 [1] и Web3.0. Подобная классификация носит концептуальный характер и используется для обозначения доминирующих архитектурных и социально-технологических особенностей различных поколений интернет-технологий. Следующий этап развития сети получил название Web3.0. В современной литературе данный термин используется для обозначения перспективной архитектуры интернет-систем, основанных на принципах децентрализации, криптографической верификации данных и распределённого управления цифровыми ресурсами. Следует отметить, что на текущий момент Web3.0 не сформировалась как полностью реализованная технологическая парадигма. В литературе выделяется набор характерных признаков, которые предположительно будут определять архитектуру будущих сетевых систем. К числу таких признаков относятся:

- децентрализация инфраструктуры;

- криптографическая верификация операций;
- использование распределённых алгоритмов консенсуса;
- интеграция интеллектуальных методов обработки данных.

Одной из ключевых технологических основ Web3.0 является блокчейн – распределённая структура данных (распределённый реестр), обеспечивающая согласованность состояния системы между множеством независимых узлов. Концепция блокчейна получила широкое распространение после публикации работы [2], в которой был предложен механизм децентрализованной электронной валюты на основе криптографических методов и распределённых алгоритмов консенсуса.

Эллиптические кривые в криптографии блокчейна. *secp256k1* – эллиптическая кривая над конечным полем F_p , задаваемая уравнением $y^2 \equiv x^3 + 7 \pmod{p}$ (см. рис. 1), где $p = 2^{256} - 2^{32} - 977$. На этой кривой зафиксирована базовая точка G порядка n [3]. Безопасность криптографических алгоритмов на эллиптических кривых обусловлена вычислительной сложностью задачи дискретного логарифмирования.

Сложение точек эллиптической кривой. Пусть есть две точки $P = (x_1, y_1), Q = (x_2, y_2)$ на кривой $y^2 = x^3 + ax + b$. Тогда вычисляется наклон прямой $\lambda = \frac{y^2 - y^1}{x^2 - x^1}$. После этого координаты новой точки: $x_3 = \lambda^2 - x_1 - x_2$ и $y_3 = \lambda(x_1 - x_3) - y_1$. Полученная точка $R = (x_3, y_3)$ и есть $R = P + Q$.

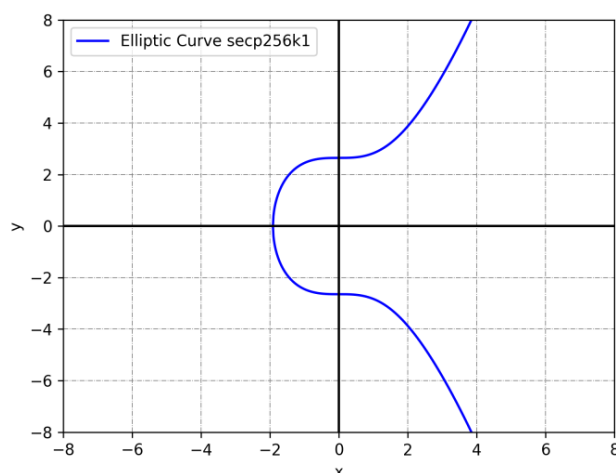


Рис. 1. secp256k1

Функционирование блокчейн-систем. С математической точки зрения функционирование блокчейна можно представить как последовательность взаимосвязанных вычислительных этапов, таких как формирование адреса, создание и подпись транзакции, проверка её

корректности узлами сети и последующее включение в блок распределённого реестра.

Создание адреса. Генерация криптографического кошелька выполняется полностью локально и не требует взаимодействия с блокчейн-сетью. До появления первой транзакции соответствующий адрес отсутствует в состоянии распределённого реестра и представляет собой лишь потенциальный идентификатор участника системы. Вся процедура выглядит так: $s \rightarrow d \rightarrow Q \rightarrow H(Q) \rightarrow A$, где:

- s – **секрет**. Обычно используется случайная битовая последовательность длины 256: $s \in \{0,1\}^{256}$. Для удобства хранения и восстановления приватного ключа широко применяется мнемоническое представление секрета – **сид-фраза**, определённая стандартом BIP-39.

- d – целочисленный **приватный ключ**. Последовательность байт секрета воспринимается как число в двоичной записи и переводится в десятичную. Должно выполняться $1 \leq d \leq n-1$, иначе генерируется новый секрет.

- Q – **публичный ключ**, который представляет собой точку $Q = (Q_x, Q_y)$ на эллиптической кривой, полученную в результате скалярного умножения базовой точки G на приватный ключ d .

- $H(Q)$ – **хеш публичного ключа**, вычисляемый по формуле $RIPMD160(SHA256(Q_x \parallel Q_y)) \in \{0,1\}^{160}$. Методики конкатенации Q_x и Q_y могут отличаться в разных алгоритмах.

- A – **адрес**, который вычисляется конкатенацией служебного байта версии, $H(Q)$ и контрольной суммы.

Формирование транзакции. Транзакцию можно представить как кортеж $T = (a_{from}, a_{to}, v, \tau)$, где a_{from} – адрес отправителя, a_{to} – адрес получателя, v – передаваемое значение, τ – служебные параметры транзакции. Транзакция представляет собой структурированное сообщение, описывающее изменение состояния распределённого реестра. Блокчейн-систему можно рассматривать как систему переходов состояния $S_{t+1} = F(S_t, T)$, где S_t – текущее состояние распределённого реестра, T – транзакция, F – функция перехода состояния.

Цифровая подпись транзакции. Цифровая подпись используется для подтверждения того, что транзакция была сформирована владельцем соответствующего криптографического кошелька и не была изменена после её создания. Для этой цели применяется алгоритм **ECDSA (Elliptic Curve Digital Signature Algorithm)** [3]. Перед подписанием транзакция сериализуется и подвергается хешированию $h = H(T)$, где T – транзакция, а H – криптографическая хеш-функция. Алгоритм формирования подписи выполняется следующим образом.

1. Генерируется случайное число $k \in [1, n-1]$.
2. Вычисляется точка $R = kG = (x_R, y_R)$.
3. Определяется значение $r = x_R \bmod n$.
4. Вычисляется значение $s = k^{-1}(h + dr) \bmod n$.

Пара значений $\sigma = (r, s)$ образует цифровую подпись транзакции.

Проверка цифровой подписи транзакции. После формирования транзакции и её подписи она распространяется по одноранговой сети. Каждый узел сети выполняет проверку корректности цифровой подписи. Проверка содержит следующие этапы.

1. Проверяется, что $1 \leq r < n$ и $1 \leq s < n$.
2. Вычисляется $w = s^{-1} \bmod n$.
3. Определяются значения $u_1 = hw \bmod n$ и $u_2 = rw \bmod n$.
4. Вычисляется точка $R(x_R, y_R) = u_1G + u_2Q$.
5. Подпись считается корректной, если выполняется условие $r \equiv x_R \bmod n$.

Если проверка успешна, это подтверждает, что транзакция была сформирована владельцем соответствующего кошелька и не была изменена после подписания. Транзакции, успешно прошедшие проверку, помещаются узлами сети в пул неподтверждённых транзакций.

Формирование блока. Узлы сети, участвующие в формировании новых блоков, выбирают набор транзакций из пула неподтверждённых транзакций и объединяют их в блок. Пусть блок содержит набор транзакций $T_1, T_2, \dots, T_m$. Каждая транзакция предварительно хешируется $h_i = H(T_i)$, $i = 1, 2, \dots, m$, где H – криптографическая хеш-функция. Для обеспечения целостности набора транзакций используется структура данных, известная как **дерево Меркла** [4]. В данной структуре хеши транзакций образуют листья бинарного дерева. Далее хеши соседних узлов попарно объединяются и снова хешируются: $h_{i,j} = H(h_i \parallel h_j)$. Этот процесс рекурсивно продолжается до получения единственного значения – **корня дерева Меркла**. Таким образом, структура блока может быть представлена в виде $B_i = (H_{i-1}, R_i, \tau_i, \nu_i)$, где H_{i-1} – хеш предыдущего блока, R_i – корень дерева Меркла, τ_i – служебные параметры блока, ν_i – параметры алгоритма консенсуса. Таким образом, дерево Меркла обеспечивает компактное представление множества транзакций и эффективную проверку целостности данных в блокчейн-системе.

Связывание блоков в цепочку. Каждый блок содержит криптографический хеш предыдущего блока. Благодаря этому блоки образуют последовательную цепочку, в которой каждый элемент

зависит от предыдущего. Изменение данных в одном из блоков приводит к изменению его хеша и, следовательно, всех последующих блоков. Такая структура называется **хеш-цепью** и обеспечивает целостность истории транзакций в блокчейн-системе.

Распределённое хранение и консенсус. Копии блокчейна хранятся на множестве независимых узлов сети. Каждый узел поддерживает собственную копию реестра и проверяет корректность новых блоков. Для согласования состояния распределённой системы используется алгоритм консенсуса **Proof-of-Work** [6]. Он позволяет узлам сети прийти к единому мнению относительно того, какой блок должен быть добавлен в цепочку следующим. Формально консенсус можно рассматривать как задачу перехода состояния системы $S_{i+1} = F(S_i, B_i)$, где S_i – текущее состояние реестра, а B_i – новый блок транзакций. Каждый узел сети самостоятельно проверяет корректность блока перед его добавлением в локальную копию цепочки. Проверка включает в себя проверку структуры блока, корректности хеша предыдущего блока, корректности цифровых подписей транзакций и соблюдения правил протокола. Если блок содержит недопустимую транзакцию или нарушает правила системы, такой блок считается невалидным и отклоняется узлами сети. Таким образом, попытка включить в блок несуществующую или некорректную транзакцию не приводит к изменению состояния распределённого реестра. Невалидный блок просто не принимается большинством узлов и не включается в основную цепочку блоков. В алгоритме Proof-of-Work узлы сети выполняют вычислительную задачу поиска такого значения *nonce*, при котором хеш блока удовлетворяет заданному условию сложности $H(B_i) < T$, где T – целевой уровень сложности. Задача консенсуса тесно связана с проблемой византийской отказоустойчивости распределённых систем. В классической постановке система из N узлов может корректно функционировать при наличии не более $f < \frac{N}{3}$ вредоносных или неисправных участников. Благодаря этому обеспечивается устойчивость системы к отказам и возможность функционирования без централизованного управляющего узла.

Заключение. Развитие технологий Web3.0 во многом является ответом на новые вызовы, возникающие в современной цифровой инфраструктуре. Одним из ключевых технологических инструментов Web3.0 является блокчейн. В архитектуре блокчейн-систем естественным образом сочетаются результаты теории чисел, криптографии, теории графов, теории вероятностей и распределённых алгоритмов. Примечательно, что данные методы, изначально разработанные в рамках фундаментальных исследований, находят

применение в решении сугубо прикладной задачи – организации надёжного, прозрачного и проверяемого механизма хранения и передачи информации без необходимости доверия к центральному посреднику.

Литература

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: <https://bitcoin.org/bitcoin.pdf> (дата обращения: 05.03.2026).
2. O'Reilly T. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. 2005. URL: <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> (дата обращения: 05.03.2026).
3. SEC 2: Recommended Elliptic Curve Domain Parameters. Version 2.0. Standards for Efficient Cryptography Group (SECG), 2010. URL: <https://www.secg.org/sec2-v2.pdf> (дата обращения: 08.03.2026).
4. Johnson D., Menezes A., Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA) // International Journal of Information Security. 2001. Vol. 1, № 1. P. 36–63.
5. Merkle R.C. Protocols for Public Key Cryptosystems // Proceedings of the IEEE Symposium on Security and Privacy. Oakland, USA, 1980. P. 122–134.
6. Gervais A. et al. On the security and performance of proof of work blockchains // Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016. C. 3-16.

Е.И. Гончаров

Smolensk State University

MATHEMATICAL ASPECTS OF WEB3.0

Keywords: *Web3.0; blockchain.*

Abstract. *The article examines the mathematical and algorithmic foundations of Web3.0 technologies with a focus on blockchain and distributed systems. The sequence of cryptographic procedures underlying blockchain operation is described, including address generation, transaction creation and signing, and the inclusion of transactions in a distributed ledger. It is shown that the implementation of these mechanisms relies on methods from several areas of mathematics, including number theory, cryptography, graph theory, and distributed algorithms.*

А.В. Городилов, А.И. Кононова

Национальный исследовательский университет «МИЭТ»

УДК 004.354

КИНЕМАТИЧЕСКАЯ МОДЕЛЬ МАНИПУЛЯТОРА

Ключевые слова: *робот, манипулятор, прямая кинематика, кинематическая модель, робототехника, планирование траектории.*

В статье представлена модель прямой кинематики многозвеного роботизированного манипулятора. Манипулятор рассматривается как последовательная кинематическая цепь, состоящая из шести суставов. Модель позволяет учесть ограничения углов суставов и настроить длины сегментов (звеньев), что позволяет адаптировать её к различным конструкциям манипуляторов. Функция прямой кинематики отображает координаты манипулятора в декартовы координаты рабочего органа и промежуточных точек сегментов. Расчёт выполняется посредством последовательного применения однородных преобразований, каждое из которых включает поворот в суставе и последующий перенос к следующему сегменту. Разработанная модель реализована в виде модуля на языке Python. Модуль также поддерживает воспроизведение команд приводов, записанных в форматах CSV и JSON, что позволяет выполнять визуализацию движения и анализ траекторий в реальном времени, подходит для планирования траекторий и предварительной проверки алгоритмов управления перед проведением экспериментов на реальных роботизированных системах.

Задача моделирования манипуляторов является актуальной, так как позволяет значительно упростить и ускорить разработку алгоритмов управления роботами и сэкономить значительные ресурсы [1]. Однако модели не являются взаимозаменяемыми, особенно если они учитывают множество специфических параметров [2]. В процессе разработки алгоритмов управления движением манипулятором по заданной траектории возникла необходимость разработки модели, учитывающей кинематические свойства и ограничения многозвеного манипулятора.

Манипулятор моделируется как последовательная цепь сегментов с шестью приводами. На рисунке 1 показана визуализация модели. В модели учитываются несколько суставов, которые в реальном роботе соответствуют серводвигателям. Каждый сустав имеет ограничение по диапазону движения, которые могут быть настроены. Длина сегмента также может быть изменена.

Назначения суставов, положение которых можно изменять в процессе моделирования:

- J1 – поворот основания вокруг вертикальной оси (yaw);
- J2–J4 – последовательные сгибания (pitch) в плоскости движения сегментов;
- J5 – вращение рабочего органа (roll) вокруг продольной оси;
- J6 – управление рабочим органом (захватом или инструментом).

Положение рабочего органа манипулятора определяется функцией прямой кинематики, которая отображает вектор углов приводов суставов $(q_1, q_2, q_3, q_4, q_5)$ в декартовы координаты точки рабочего органа. Функция прямой кинематики реализует отображение из пространства суставных координат в декартово пространство рабочего органа. На вход функция получает вектор углов приводов $(q_1, q_2, q_3, q_4, q_5)$ и возвращает положение характерных точек манипулятора, включая рабочий орган. В минимальной постановке задачи, необходимой для задач планирования траектории, результатом является радиус-вектор центра рабочего органа $p = (x, y, z)$. Модель также принимает и хранит шестую координату – угол привода рабочего органа (q_6), которая не влияет на декартовы координаты характеристических точек или ориентацию суставов, но используется в визуализации и может быть получена из модели.

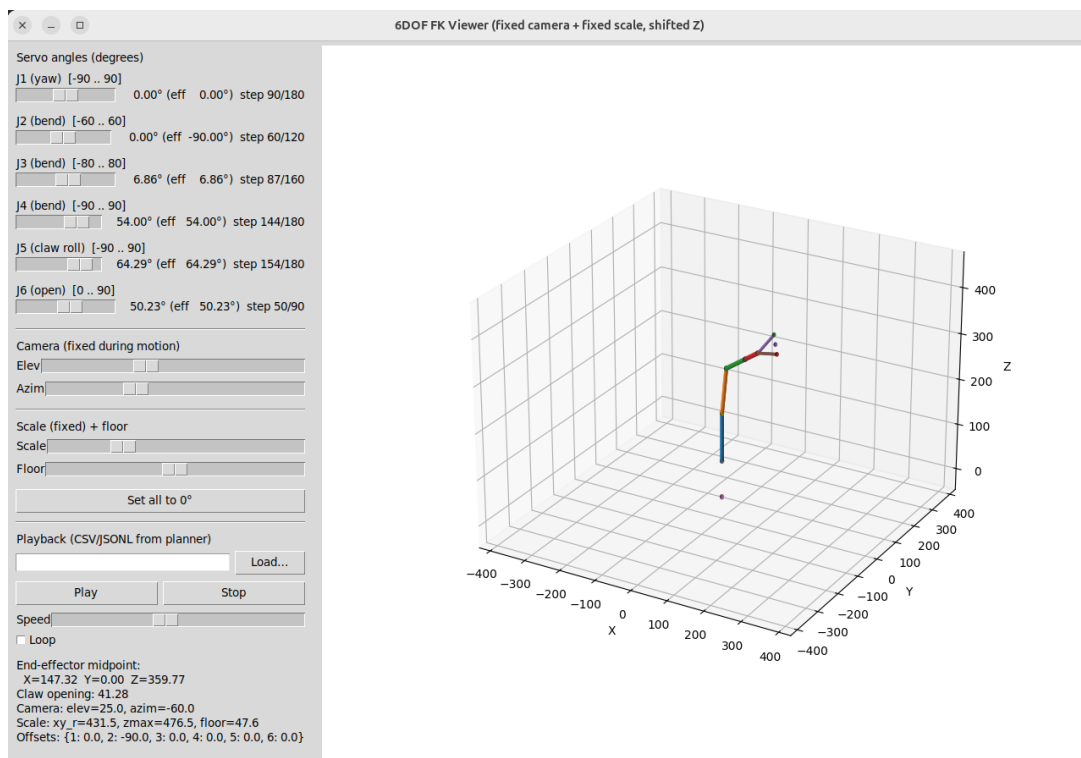


Рис.1. Визуализация кинематической модели манипулятора

Расчёт выполняется посредством последовательного применения однородных преобразований, каждое из которых состоит из поворота в суставе и переноса к следующему сегменту. Перенос на вектор (x, y, z) задаётся матрицей:

$$T_i(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \dots$$

Вращение вокруг осей также описывается матрицами:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Итоговое преобразование от базы к рабочему органу задаётся произведением:

$$T = R_z(q_1) \cdot T_t(0,0,h) \cdot R_y(q_2 + \theta_{\text{off}}) \cdot T_t(l_2,0,0) R_y(q_3) \cdot T_t(l_3,0,0) \cdot \\ \cdot R_y(q_4) \cdot T_t(l_4,0,0) \cdot R_x(q_5) \cdot T_t(l_{\text{tool}},0,0),$$

где h – высота основания, l_2, l_3, l_4 – длины сегментов, l_{tool} – длина рабочего органа, θ_{off} – постоянное смещение второго сустава.

Таким образом, каждый сегмент описывается локальной системой координат, начало которой совпадает с точкой вращения соответствующего сустава. Переход к следующему сегменту выполняется последовательным применением поворота в суставе (матрицы R_x, R_y, R_z) и переноса вдоль локальной оси X (матрица T_t) на длину сегмента l_i . Поскольку матрицы преобразования умножаются последовательно, ориентация каждого следующего сегмента автоматически учитывает все предыдущие повороты. Координаты конца каждого сегмента определяются как положение начала его локальной системы координат после применения накопленного преобразования.

Координаты рабочего органа извлекаются из последнего столбца результирующей матрицы: $p = (T_{14}, T_{24}, T_{34})$. При этом всегда $T_{44} = 1$.

Для удобства интерпретации результатов к углам прибавляются постоянные смещения, равные половине диапазона каждого угла. Таким образом, при установке всех углов в 0 манипулятор занимает вертикальное положение.

Модель реализована как модуль на языке Python. Программный интерфейс модуля позволяет задавать текущее положение манипулятора в координатах манипулятора и получать положение конца каждого сегмента в декартовых координатах. Ограничения по углам и длины сегментов могут быть изменены для того, чтобы соответствовать конкретному манипулятору. Модуль поддерживает воспроизведение сохраненных команд сервоприводов в формате CSV и JSON, что

позволяет увидеть и оценить движения манипулятора в реальном времени.

Данная модель активно применяется в исследованиях, связанных с роботами-манипуляторами в СПИНТех МИЭТ. Её использование позволяет планировать эксперименты с реальными роботами, проверять алгоритмы управления и обнаруживать возможные опасные движения. Сама модель и некоторые примеры её применения доступны для загрузки и использования на GitHub [3].

Модель является чисто кинематической: не учитываются динамика, ограничения по моментам, упругие деформации и столкновения. Предполагается, что команды приводов исполняются идеально в заданных диапазонах.

Для построения траектории и многих других задач данные допущения являются приемлемыми, так как игнорируемые параметры определяются материалами и качеством самих приводов. При этом модель учитывает принципиальные ограничения. Построение траектории в рамках ограничений этой модели позволяет избежать решений, реализация которых невозможна на практике.

Литература

1. A set of regression models for controlling a trolley equipped with a robot manipulator integrated into the technological process / I. Yu. Kholodilin, M. A. Grigorev, V. A. Kushnarev [et al.] // Bulletin of South Ural State University. Series: Power Engineering. 2024. Vol. 24, No. 4. P. 47-55.

2. Попиков П.И., Клубничкин В.Е. Математическая модель функционирования энергосберегающего гидропривода механизма поворота лесного манипулятора // Лесотехнический журнал. 2015. Т. 5, № 4(20). С. 223-234.

3. RobotMovementModel, URL: <https://github.com/lexiehexie/RobotMovementModel> (дата обращения: 02.03.2026).

A.V. Gorodilov, A.I. Kononova

National Research University of Electronic Technology (MIET)

KINEMATIC MODEL OF A ROBOTIC MANIPULATOR

Keywords: *robot, manipulator, forward kinematics, kinematic model, robotics, trajectory planning.*

Abstract: *This paper presents a forward kinematics model of a multi-link robotic manipulator. The manipulator is represented as a serial chain consisting of six actuated joints. The model takes into account joint angle limits and configurable link lengths, allowing adaptation to different manipulator designs. The forward kinematics function maps joint-space coordinates to Cartesian coordinates of the end-effector and intermediate link points. The computation is performed through sequential application of*

homogeneous transformations, each consisting of a joint rotation followed by a translation to the next link. The developed model is implemented as a Python module and provides an interface for setting joint angles and obtaining Cartesian positions of all manipulator segments. The module also supports playback of recorded actuator commands in CSV and JSON formats, enabling real-time visualization and motion analysis. The model is purely kinematic and does not account for dynamics, torque limits, elastic deformations, or collision detection. Nevertheless, it is suitable for trajectory planning and preliminary validation of control algorithms before experiments on real robotic systems.

В.Н. Захаров

Федеральный исследовательский центр «Информатика и управление» Российской академии наук

УДК 004.3

ВИРТУАЛЬНЫЙ КОМПЬЮТЕРНЫЙ МУЗЕЙ КАК ИСТОЧНИК ЗНАНИЙ ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

Ключевые слова: виртуальный компьютерный музей; ЭВМ; совет музея; вычислительная техника; компонентная база; галерея славы.

В статье представлены сведения о Российском виртуальном компьютерном музее: истории его создания, основных создателей музея, его структуре и содержании информационных разделов. Описывается деятельность Совета музея по пополнению его содержания, приводятся данные о проводимых мероприятиях, в частности о конференциях по истории развития вычислительной техники в России, странах бывшего СССР и СЭВ. Говорится об использовании материалов музея.

1. Введение. История создания компьютеров в СССР восходит к концу 1940-х годов, когда был получен первый отечественный патент И.С. Бруком и Б.И. Рамеевым на структуру цифровой вычислительной машины (приоритет от 4 декабря 1948 года), и к началу 1950-х годов, когда были созданы первые отечественные ЭВМ М1 (под руководством И.С. Брука) и МЭСМ (под руководством С.А. Лебедева). В академических и промышленных организациях достаточно интенсивно разрабатывались новые ЭВМ, программное обеспечение для них,

строились заводы по производству средств вычислительной техники [1]. С распадом СССР в 1991 году ситуация с созданием отечественной вычислительной техники резко ухудшилась. Возобладала политика ориентации на приобретение зарубежных ЭВМ. Ряду людей, принимавших участие в создании и производстве отечественных средств вычислительной техники, на фоне происходящих событий возникло опасение, что богатая история развития этой важнейшей отрасли может быть потеряна.

Автором идеи о создании Российского виртуального компьютерного музея был Эдуард Пройдаков, в то время главный редактор российского еженедельника «PCWeek-RE», который 30 сентября 1997 года в редакционной колонке главного редактора опубликовал свою статью «Давайте создадим музей» [2]. В этой статье содержался следующий основополагающий призыв: «Время, к сожалению, неумолимо. Уходят из жизни люди, судьбы которых связаны с историей отечественных разработок. Но что знают о них наши дети и внуки? Я пришёл к выводу, что нам совершенно необходимо собрать, насколько это возможно, следы уходящих компьютерных поколений и организовать музей вычислительной техники. Обращаясь к коллегам, я прошу создание Виртуального компьютерного музея (ВКМ) рассматривать не как очередную акцию газеты «PC Week/RE», а как нашу общую важную культурную миссию, служащую для исполнения долга перед будущими поколениями читателей. Создание ВКМ является тем делом, которое можно сделать, лишь объединив усилия».

2. Начальные шаги. Зимой 1998 года были предприняты первые практические шаги по реализации предложенной идеи создания ВКМ. Прежде всего был образован Совет ВКМ, в который дали согласие войти ведущие разработчики советских ЭВМ. Первым председателем Совета ВКМ стал Виктор Владимирович Пржиялковский, Герой Социалистического Труда, который в течение ряда лет был директором НИЦЭВТ, генеральным конструктором ЭВМ Единой серии (ЕС ЭВМ). Он очень много сделал на первом (организационном) этапе создания ВКМ. Заместителем председателя Совета ВКМ стал Евгений Николаевич Филинов, работавший ещё под руководством И.С. Брука в Энергетическом институте АН СССР, затем заместителем директора ИНЭУМ, заместителем директора ИПИ АН, заместителем директора ИСП РАН. Также в состав Совета ВКМ вошли известные конструкторы советских компьютеров Ярослав Афанасьевич Хетагуров, Николай Петрович Брусенцов, Юрий Васильевич Рогачёв, Виталий Иосифович Штейнберг, ветераны создания советских компьютеров Тамара Миновна Александриди, Александр Николаевич Томилин. Позднее в состав Совета входили специалисты - энтузиасты из различных организаций (в том числе НИИ, предприятий, вузов, музеев),

работавшие в области вычислительной техники и информатики. Бессменным директором и главной движущей силой ВКМ является Эдуард Михайлович Пройдаков, а ответственным секретарем, ведущим значительную организационную работу, – Наталья Ариановна Черемных.

В.В. Пржиялковский и Е.Н. Филинов возглавили работу по определению структуры ВКМ и подготовке первых материалов по семействам компьютеров. Ими же было написано несколько биографий выдающихся ученых и конструкторов для задумывавшегося с самого начала раздела «Галерея славы». По мере роста объема накопленных материалов росло количество структурных разделов ВКМ.

В настоящее время Виртуальный компьютерный музей представляет собой крупный портал, содержащий разделы на русском и английском языках. Остановимся на содержании русскоязычной части ВКМ. На главной странице написано: «Идея создания нашего музея была простой и бесхитростной – мы хотели собрать, систематизировать и обнародовать банк данных, связанный с происхождением и развитием вычислительной техники, прежде всего отечественной. Нами двигало стремление не дать кануть в Лету уникальным материалам, способным осветить более чем полувековую историю отечественного компьютеростроения, его взлет и падение. Но мы ни в коей мере не хотим брать на себя роль судей. Мы претендуем только на роль добросовестных летописцев» [3].

3. Структура и содержание ВКМ. Главная страница состоит из трех колонок. Центральная содержит регулярно пополняемые новости, в том числе оперативные сведения о размещении дополнительных материалов в ВКМ. В настоящее время раздел новостей занимает 89 страниц и содержит все сообщения, начиная с 2013 года.

В левой колонке приведены ссылки на информационные блоки: «О нашем музее», «История отечественной вычислительной техники», «История управляющей вычислительной техники», «История развития электросвязи», «История отечественной электронной компонентной базы (ЭКБ)», «Галерея славы», «Документы и публикации», «Книги и компьютерная пресса», «Англо-русский компьютерный словарь». В правой колонке даны ссылки на информационные блоки: «Вычисления в докомпьютерную эпоху», «История вычислительной техники за рубежом», «История развития программного обеспечения», «Технологии», «Календарь событий», «История компьютерных игр», «Музеи И ВТ в Интернет», «Архив новостей», «Гостевая книга». Каждый из 18 блоков, в свою очередь, состоит из многочисленных подразделов, часто разделенных на большее количество рубрик.

Информацию о членах Совета ВКМ, контактные адреса и информацию для партнеров музея посетители сайта могут найти в информационном блоке «О нашем музее».

Информационный раздел «История отечественной вычислительной техники» содержит материалы о развитии советских компьютеров. Он состоит из девяти крупных подразделов:

- «Машины по алфавитному указателю» с информацией о 346 советских компьютерах, каждое название машины является ссылкой на размещенный материал о соответствующем компьютере;

- «Первые ЭВМ» с 12 подразделами, содержащими статьи и некоторые делящиеся на дополнительные подразделы. В нем представлены первые советские электронные ламповые компьютеры, такие как проекты, инициированные и руководимые пионерами советской вычислительной техники, выдающимися учеными И.С. Бруком (компьютеры «М-1», «М-2», «М-3»), С.А. Лебедевым (компьютеры «МЭСМ» и семейство «БЭСМ»), Б.И. Рамеевым «Урал» и Ю.Я. Базилевским (первый советский серийный компьютер «Стрела»);

- «Специализированные ЭВМ военного назначения», состоящий из 36 статей, содержит подробные описания специальных компьютеров, используемых в советских системах противовоздушной и противоракетной обороны, военно-морских компьютеров, используемых на боевых кораблях, бортовых компьютеров военно-воздушных сил, компьютеров для ракетных войск стратегического назначения, военно-космических систем и т.д. Этот подраздел завершается списком советских военных компьютеров;

- «Управляющие ЭВМ», содержащий 15 статей и подразделов. В нем представлена информация о компьютерах и вычислительных комплексах для управления промышленными предприятиями и контроля промышленных технологических процессов, выпущенных в период 1960–1980-х годов;

- «Универсальные ЭВМ», содержащий 22 статьи. Представляет собой большой информационный кластер о семействе компьютеров «БЭСМ», большом семействе компьютеров «УРАЛ», многочисленных компьютерах серии ЕС ЭВМ, выпускавшихся в Армении компьютерах «Арагац», «Раздан», «Наири»; уникальных троичных компьютерах «Сетунь»; выпускавшихся на Украине компьютерах «Киев», «Проминь», «Мир»; семействе суперкомпьютеров «Эльбрус»; космических / военных компьютерах «Весна» и «Снег»; крупной серии компьютеров «Минск», производимых в Белоруссии, и многих других;

- «Аналоговые ЭВМ», содержащий 8 статей. Характерными чертами этих устройств были простота синхронизации с различными реальными устройствами и приборами, простота настройки и переналадки для решения различных задач и удобства взаимодействия с пользователями. Аналоговые компьютеры, разработанные в 1930-х годах компьютерными пионерами С.А. Лебедевым, И.С. Бруком, Л.И.

Гутенмахером, были известны, популярны и широко внедрялись в СССР, в первую очередь в электроэнергетике;

- «История развития полупроводниковой промышленности», содержащий три большие статьи;

- «Системы и комплексы», содержащий 18 статей;

- «Юбилеи», содержащий материалы к юбилейным датам известных людей и событий, в настоящее время содержит 31 статью.

Раздел «История управляющей вычислительной техники» содержит подразделы «Автоматизированные информационно-управляющие системы», «Компьютерные технологии для встраиваемых систем», «ПО и ОС для систем реального времени», «Устройства связи с объектом», «Разработчики», «Производители». При этом структура предполагает, что подразделы регулярно пополняются.

Раздел «История развития электросвязи» является одним из крупнейших в составе ВКМ. Он содержит подразделы «Очерки развития электросвязи», «Проводная связь», «Радиосвязь», «Телевидение и радиовещание» и по сути является «музеем в музее».

Раздел «История отечественной электронной компонентной базы (ЭКБ)» содержит подразделы «История электроники», «Предприятия электронной промышленности», «Ламповая электроника».

Раздел ВКМ «Галерея славы», наряду с «Историей отечественной вычислительной техники», является одним из самых важных и крупных разделов. «Галерея славы», в свою очередь, состоит из подразделов «Отечественные ученые и инженеры» и «Зарубежные ученые и инженеры», которые содержат биографические статьи о 107 советских ученых, а также 25 статей о таких всемирно известных личностях, как А. Тьюринг, К. Цузе, В. Буш, С. Макнили, Н. Вирт, Р. Хэмминг и С. Джобс (соучредитель Apple Inc.).

Раздел «Документы и публикации» содержит подразделы «Материалы конференций», содержащий материалы 16 конференций, включая все шесть проведенных международных конференций по истории вычислительной техники SoRuCom; «Законодательные акты и постановления»; «Нормативные и методические документы».

Раздел «Книги и компьютерная пресса» содержит подразделы «Электронная библиотека» (сюда входит 195 оцифрованных книг, относящихся к истории информатики и вычислительной техники), «Библиографический справочник», «Ведущие компьютерные издания», «Игровая пресса».

В разделе «Англо-русский компьютерный словарь» размещена ИТ-энциклопедия, содержащая, как указано в настоящий момент, 11 029 статей, составленная Э.М. Пройдаковым и Л.А. Теплицким.

4. Проводимые конференции. Совет ВКМ совместно с Российским политехническим музеем, рядом ведущих университетов и

других научных центров (ранее при поддержке Российского фонда фундаментальных исследований) регулярно организует всероссийские, а также международные конференции, посвященные истории создания и практического применения компьютеров, компьютерных и программных комплексов и систем в СССР и в настоящее время в России. Было проведено шесть международных конференций:

I Международная конференция «Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы (SoRuCom-2006)», Петрозаводск;

II Международная конференция «Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР (SoRuCom-2011)», Великий Новгород;

III Международная конференция «Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР: история и перспективы (SoRuCom-2014)», Казань;

IV Международная конференция «Развитие вычислительной техники и её программного обеспечения в России и странах бывшего СССР: история и перспективы (SoRuCom-2017)», Зеленоград;

V Международная конференция «Развитие вычислительной техники в России, странах бывшего СССР и СЭВ (SoRuCom-2020)», Москва;

VI Международная конференция «Развитие вычислительной техники в России, странах бывшего СССР и СЭВ (SoRuCom-2023)», Нижний Новгород.

В сентябре 2026 года в Ярославле состоится VII Международная конференция «Развитие вычислительной техники в России, странах бывшего СССР и СЭВ (SoRuCom-2026)».

5. Деятельность Совета ВКМ. Члены Совета VCM постоянно совершенствуют структуру музея, чтобы сделать его максимально интересным и удобным для посетителей. После Виктора Васильевича Пржиялковского Совет музея возглавляли Ярослав Афанасьевич Хетагуров, затем Виталий Иосифович Штейнберг. В настоящее время председателем Совета ВКМ является многолетний учёный секретарь Института проблем информатики АН СССР (позднее ИПИ РАН и Федеральный исследовательский центр «Информатика и управление» РАН) Виктор Николаевич Захаров, а его заместителями являются Геннадий Алексеевич Егоров (бывший заместитель директора ИНЭУМ) и Владимир Анатольевич Китов (доцент РЭУ им. Г.В. Плеханова).

Многочисленные книги, научные и иные доклады, тезисы и материалы различных компьютерных конференций, соответствующие директивы советского правительства и многое другое, что можно было бы отнести к областям «информатика» и «информационные технологии», переводятся в цифровую форму и помещаются в

соответствующие разделы музея. В настоящее время ВКМ размещает на своем сайте (в открытом доступе) от 10 до 15 архивных публикаций в месяц. В структуре музея размещено и хранится более 3000 исторических документов и статей (как архивных, так и написанных специально для ВКМ). Его электронная библиотека содержит более 150 научных и учебных книг по информационным технологиям (изданных в 1950–1960-х годах), которые трудно найти в других источниках [4].

6. Заключение. Согласно статистике, которая составляется на основе данных счетчиков посещений музея, от одной до двух тысяч человек из разных стран ежедневно работают с материалами ВКМ, переходя на его сайты по ссылкам, предоставляемым поисковыми системами. Количество посетителей, естественно, зависит от текущего сезона года или периода обучения в школах и университетах. Таким образом, оно резко возрастает во время подготовки семестровых проектов студентами университетов, поскольку ВКМ рекомендован российскими образовательными организациями в качестве вспомогательного информационного ресурса для изучения информатики. Таким образом, студенты составляют одну из основных групп пользователей ВКМ.

Материалы музея одинаково интересны специалистам как в области информационных технологий и их применения, так и в области вычислительной техники. Особый интерес к публикациям музея проявляют ветераны вычислительной техники, ученые, программисты, компьютерные дизайнеры и другие люди, чья профессиональная деятельность была или остается связанной с разработкой и применением компьютеров.

Литература

1. Захаров В.Н. К истории создания и производства ЭВМ в СССР // Вестник РАН. 2025. № 12. С. 56–69.
2. Пройдаков Э.М. Давайте создадим музей. URL: www.pcweek.ru/themes/detail.php?ID=43652.
3. Виртуальный компьютерный музей. computer-museum.ru.
4. Vladimir A. Kitov, Edward M. Proydakov. TWENTIETH ANNIVERSARY OF THE RUSSIAN VIRTUAL MUSEUM OF THE COMPUTERS AND INFORMATION TECHNOLOGY HISTORY. URL: https://link.springer.com/chapter/10.1007/978-3-030-29160-0_15.

V.N. Zakharov

*Federal Research Center «Computer Science and Control» of the
Russian Academy of Sciences (FRC CSC RAS)*

VIRTUAL COMPUTER MUSEUM AS A SOURCE OF KNOWLEDGE ON COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY

Keywords: *virtual computer museum, computers, museum council, computing technology, component base, hall of fame.*

Abstract. *The article presents information about the Russian Virtual Computer Museum: its history, founders, structure, and the contents of its information sections. It describes the Museum Council's efforts to expand its content and provides information on events held, including conferences on the history of computing in Russia, the former Soviet Union, and the Council for Mutual Economic Assistance (CMEA). The use of museum materials is also discussed.*

И.Г. Игнатова, М.В. Слюсарь
Национальный исследовательский университет «МИЭТ»

УДК 004.67

ОБНАРУЖЕНИЕ И КЛАССИФИКАЦИЯ НЕИСПРАВНОСТЕЙ В ЭЛЕКТРИЧЕСКИХ СЕТЯХ НА ОСНОВЕ АЛГОРИТМА БАЙЕСА–СДКП–САМОВНИМАНИЯ

Ключевые слова: *линии электропередачи, байесовская оптимизация, сети долговременной кратковременной памяти, механизм внимания, обнаружение и классификация неисправностей.*

Линии электропередачи, являясь важнейшим компонентом энергосистемы, играют значительную роль в обеспечении безопасной и стабильной работы электросети. Для решения задачи точной характеристики сложных и разнообразных типов повреждений в статье предлагается метод обнаружения и классификации повреждений линий электропередачи, который объединяет байесовское рассуждение, сети долгой краткосрочной памяти (СДКП) и механизм самовнимания. Байесовское рассуждение используется для настройки гиперпараметров СДКП, в то время как сеть СДКП эффективно обрабатывает последовательные данные с помощью своего механизма управления. Механизм самовнимания адаптивно присваивает веса, фокусируясь на взаимосвязях между информацией в разных позициях последовательности, улавливая глобальные зависимости. Результаты тестирования показывают, что предложенная модель Байеса–СДКП–

Самовнимания, демонстрирует сбалансированную и надежную работу при классификации неисправностей.

1. Введение

Надежность энергосистем критически зависит от линий электропередачи, которые часто выходят из строя. Традиционные методы обнаружения неисправностей устарели, а интеллектуальные методы (глубокое обучение) сталкиваются с проблемами обработки данных и неопределенности. Это делает насущной проблему разработки новых, сбалансированных методов.

В данной работе обнаружение неисправностей рассматривается как классификация обнаруженных линий. Существующие методы диагностики неисправностей включают метод опорных векторов, искусственные нейронные сети и сети долгой краткосрочной памяти (СДКП). Метод опорных векторов широко применяется для обнаружения неисправностей в линиях электропередачи благодаря своей превосходной способности классифицировать данные малого размера и высокой размерности. Этот метод хорошо работает при классификации данных малого размера и низкой размерности, но его эффективность зависит от выбора ядерных функций и параметров регуляризации.

СДКП, как особый тип рекуррентной нейронной сети (РНС), может эффективно обрабатывать данные временных рядов и широко используется для обнаружения неисправностей в линиях электропередачи.

В данной работе предложен новый метод обнаружения и классификации повреждений линий электропередачи, интегрирующий байесовское рассуждение (БР) для снижения неопределенности, сети долговременной кратковременной памяти для моделирования временных рядов и механизм самовнимания для фокусировки на ключевой информации.

2. Материалы и методы

2.1. Модель Байеса–СДКП–Самовнимания на основе механизма самовнимания для классификации повреждений линий электропередачи

Эффективность обучения СДКП зависит от гиперпараметров. Байесовская оптимизация как метод глобальной оптимизации требует меньше вычислений, аппроксимируя сложные функции с минимальными предположениями о гладкости.

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}. \quad (1)$$

В последнем уравнении $p(A)$ представляет собой априорное распределение, определяющее распределение суррогатной модели, $p(B)$ обозначает распределение наблюдаемых данных B , пока $p(B|A)$ представляет распределение наблюдаемых B . Апостериорное

распределение $p(A|B)$, представляет собой обновленное распределение аппроксимирующей модели после включения наблюдаемых данных B .

Долговременная кратковременная память

СДКП – это тип нейронной сети, решающий проблемы градиента и памяти рекуррентных нейронных сетей с помощью сложной четырехслойной архитектуры (входной, забывающий, выходной вентили и ячейка памяти). Сигмоидная функция σ и Tanh используются для регулирования состояний вентилях: забывания, входного и выходного.

Механизм самовнимания

Механизм самовнимания адаптивно присваивает веса, фокусируясь на взаимосвязях между информацией в разных позициях входной последовательности, тем самым улавливая глобальные зависимости в данных последовательности.

Расчет весовых коэффициентов внимания

Сходство между векторами запроса и ключа вычисляется с помощью скалярного произведения, после чего выполняется нормализация:

$$A = \text{soft max} \frac{QKT}{\sqrt{dk}}. \quad (2)$$

Вычисление взвешенных векторов значений

Матрица весовых коэффициентов внимания A используется для вычисления взвешенной суммы векторов значений W , генерируя окончательный результат:

$$Wo=AV. \quad (3)$$

Метрики оценки классификации

Матрица ошибок используется для сравнения прогнозируемых значений с фактическими значениями для четырех типов повреждений линий электропередачи, что позволяет детально отобразить распределение правильных и неправильных результатов классификации.

Для задачи бинарной классификации матрица ошибок представлена в таблице 1.

Таблица 1

Форма матрицы ошибок для задач бинарной классификации

	Прогнозируется как положительный класс	Прогнозируется как отрицательный класс
Фактически классифицируется как положительный	Истинноположительный	Ложноотрицательный
Фактически классифицируется как отрицательный	Ложноположительный	Истинноотрицательный

ИП (истинно положительный): фактический положительный класс, прогнозируемый как положительный. ЛО (ложноотрицательный): Фактический положительный класс, прогнозируемый как отрицательный. ЛП (ложноположительный): фактический отрицательный класс, прогнозируемый как положительный. ИО (истинно отрицательный): фактический отрицательный класс, прогнозируемый как отрицательный.

Для оценки результатов обучения этого алгоритма используются как метрика площади полигона (ROC), так и кривая рабочих характеристик приемника (AUC).

В алгоритме ROC в качестве вершин многоугольника выбираются следующие шесть метрик эффективности классификации.

Точность классификации (ТК):

$$CA = \frac{TP + TN}{TP + TN + FP + FN}. \quad (4)$$

Отражает долю образцов, для которых модель дает правильные прогнозы, из всех образцов, участвующих в прогнозировании.

Чувствительность:

$$SE = \frac{TP}{TP + FN}. \quad (5)$$

Чувствительность модели к образцам положительного класса выше, чем выше чувствительность, тем ниже вероятность пропуска обнаружения.

Специфичность (СП):

$$SE = \frac{TP}{TP + FP}. \quad (6)$$

Коэффициент Каппа (K):

$$K = \frac{TP}{TP + FP + FN}. \quad (7)$$

Сравнивает прогностическую эффективность модели с результатами случайной классификации.

Показатель F1 (FM):

$$Pr = \frac{TP}{TP + FP}, \quad (8)$$

$$Re = \frac{TP}{TP + FN}, \quad (9)$$

$$FM = \frac{2Pr * Re}{Pr + Re}, \quad (10)$$

Кривая AUC иллюстрирует взаимосвязь между истинно положительными показателями (ИП) и ложноположительными показателями (ЛП) при различных значениях порога.

Ось X:

$$FPR = \frac{FP}{FP + TN}. \quad (11)$$

Отражает долю отрицательных образцов, которые ошибочно классифицируются как положительные.

Ось Y:

$$TPR = \frac{TP}{TP + FN}. \quad (12)$$

Отражает долю положительных образцов, которые были правильно классифицированы.

В данном исследовании в качестве основного инструмента оценки эффективности моделей классификации использовалась AUC-кривая, а для количественного анализа моделей – метрика ППК.

2.2. Процесс классификации неисправностей

Предложенная в данной статье модель объединяет байесовский вывод, СДКП и механизм самовнимания для классификации повреждений линий электропередачи. Сохраняя исходные сигналы о повреждениях, она позволяет избежать вмешательства человека в процесс предварительной обработки данных, тем самым поддерживая их качество.

Модель классификации неисправностей имеет последовательную структуру, состоящую из входного слоя (Input), слоя СДКП (LSTM, Long-Shot Term Memory), слоя механизма самовнимания (Self Attention), слоя Dropout, полносвязного слоя (Fully Connected) и слоя Softmax. Конечный выходной сигнал генерируется с помощью функции кросс-энтропии softmax.

Входной слой модели получает последовательные, слой СДКП захватывает временные характеристики и выдает как скрытое состояние, так и состояние ячейки, затем к выходу СДКП применяется функция активации ReLU (выпрямленный линейный блок) для внесения нелинейности и повышения репрезентативных возможностей модели. Для дальнейшего учета зависимостей между различными позициями в последовательности в слой самовнимания встроен механизм самовнимания. Этот слой использует матрицы весов WQ , WK , и WV . Для вычисления взвешенной суммы выходных данных СДКП генерируется карта внимания. Во избежание переобучения используется слой Dropout с заданным коэффициентом отсечения. Полносвязный слой интегрирует извлеченные признаки, создавая выходные данные размерностью 4. Слой Softmax вычисляет вероятности классификации, генерируя распределение вероятностей по четырем классам, а функция потерь кросс-энтропии количественно оценивает ошибку классификации.

В процессе обучения функция потерь непрерывно корректирует веса и смещения, чтобы минимизировать разницу между выходным и желаемым значением. В данной работе используется функция потерь категориальной кросс-энтропии, которая вычисляется следующим образом:

$$L = -\frac{1}{n} \sum_{i=1}^n x_i \log x_i. \quad (13)$$

В уравнении L – функция потерь, $\hat{x} = \{x_1, x_2, \dots, x_n\}$ – это прогнозируемое значение, и $x = \{x_1, x_2, \dots, x_n\}$ – истинное значение.

3. Результаты

Входные данные и настройка параметров

В работе набор данных состоит из 357 образцов (249 образцов для обучающей выборки и 108 – для тестовой выборки) с 12 входными признаками. Признаки 1 и 2 отражают базовые условия работы системы, признаки 3–6 представляют собой измерения в установившемся режиме, такие как фазные токи и мощность, признаки 7 и 8 указывают на показатели качества, такие как гармоники и коэффициент мощности, а признаки 9–12 представляют собой высокочастотные переходные процессы и компоненты последовательности.

Анализ результатов классификации

На рисунке 1 показана взаимосвязь между точностью обучения, функцией потерь и количеством итераций для диагностики неисправностей с использованием модели Байеса–СДКП–Самовнимания.

Кривые точности и функции потерь показывают, что модель быстро сходится в течение первых 50 итераций. Функция потерь быстро уменьшается, в то время как точность резко возрастает, что указывает на то, что модель эффективно улавливает большинство особенностей дефектов линий на ранних этапах обучения.

Классификация повреждений линий электропередачи в данной работе включает четыре категории, образуя задачу классификации по четырем классам. Для исследования были выбраны четыре типа повреждений: однофазные замыкания на землю, двухфазные короткие замыкания, двухфазные замыкания на землю и трехфазные короткие замыкания. В таблице 2 показано распределение классов в наборе данных о повреждениях, и набор данных гарантирует, что различия в доле между категориями составляют менее 1% благодаря «стратифицированной выборке», без существенных проблем с дисбалансом классов. В таблице 3 представлены результаты прогнозирования для обучающего и тестового наборов данных.

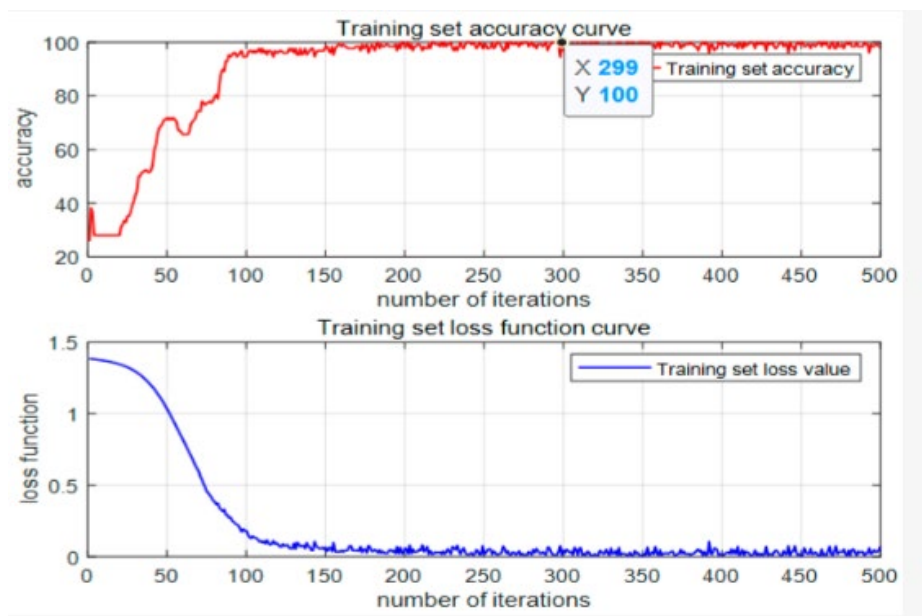


Рис. 1. Кривая точности (ассурацы) и функции потерь (Loss Function) на обучающем наборе данных

Таблица 2

Распределение классов наборов данных о неисправностях

Категория неисправности	Количество обучающих выборок	Количество тестовых выборок	Общее количество выборок	Пропорция
Тип 1	62	27	89	24.93%
Тип 2	62	27	89	24.93%
Тип 3	62	27	89	24.93%
Тип 1	63	27	90	25.21%
Общее	249	108	357	100%

Таблица 3

Результаты прогнозирования на обучающем и тестовом наборах данных

Категория выборки	Прогноз правильный	Ошибка прогноза	Точность
Тренировочный набор	239	10	95,9%
Тестовый набор	103	5	95,4%

На рисунке 2 показана метрика оценки площади многоугольника для прогнозов классификации неисправностей на тестовом наборе данных, а на рисунке 3 представлена ROC-кривая для этого набора. Пунктирная линия представляет собой идеальную границу классификации, а заштрихованная область – диапазон производительности модели на тестовом наборе. В целом эти оценочные показатели высоки, при этом значение РАМ составляет 0,92, что указывает на то, что модель Байес–СДКП–Самовнимание обеспечивает

сбалансированную и надежную классификацию повреждений линий электропередач.

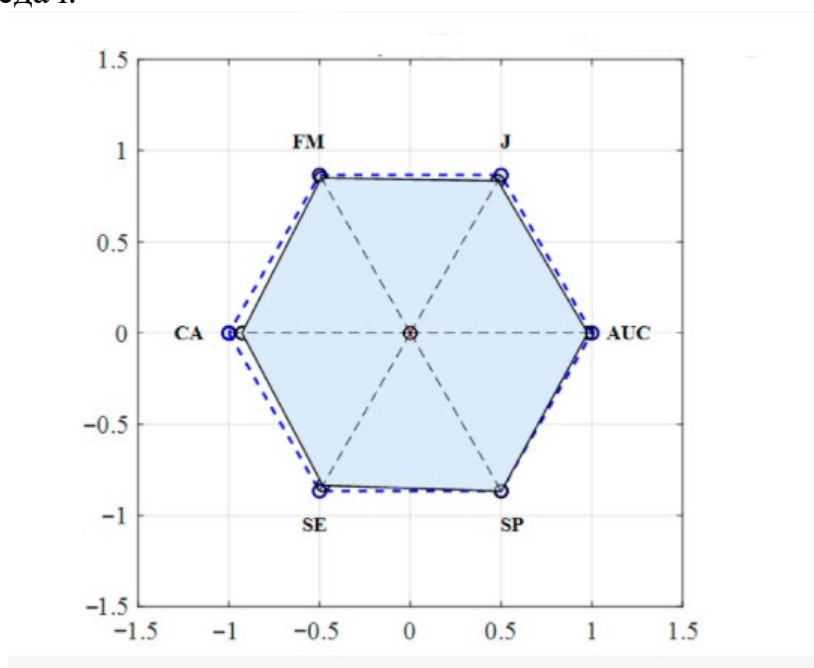


Рис. 2. Диаграмма оценки площади полигональной области тестового набора для классификации неисправностей

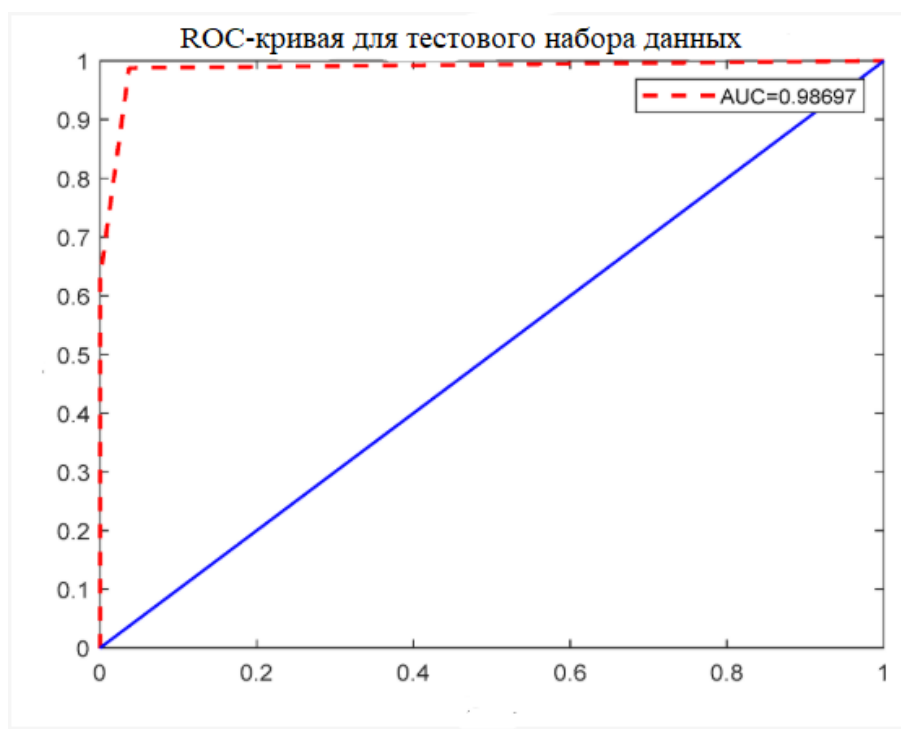


Рис. 3. Результаты ROC-кривой для тестового набора данных по классификации повреждений линий электропередач

Кривая ROC практически точно совпадает с верхним левым углом, что указывает на сохранение моделью высокой способности к различению неисправностей при различных пороговых значениях.

4. Выводы

В данной статье предлагается модель классификации повреждений трехфазных линий электропередачи, которая сочетает в себе байесовскую оптимизацию, механизмы самовнимания и сети СДКП. Метрики оценки рассчитываются с использованием метода оценки площади полигона, что приводит к следующим выводам.

1. Благодаря изучению 12 различных характеристик образцов неисправностей с помощью сети СДКП предложенная модель Байеса–СДКП–Самовнимания демонстрирует высокую точность классификации неисправностей.

2. Благодаря внедрению механизма самовнимания модель способна адаптивно фокусироваться на ключевой информации в последовательности неисправностей, повышая свою способность к анализу глобальных зависимостей.

Данный метод обеспечивает эффективную поддержку эксплуатации и технического обслуживания линий электропередачи, а модель на основе глубокого обучения имеет потенциал для адаптации к различным динамическим сценариям в практических приложениях. Дальнейшие исследования могли бы быть сосредоточены на оптимизации структуры модели или включении дополнительных характеристик неисправностей для повышения точности классификации и решения более сложных задач.

Литература

1. Быкадоров А.Л., Муратова-Милехина А.С. Метод определения места короткого замыкания в тяговых сетях переменного тока на основе информационных технологий // Труды Междунар. научно-практ. конф. «Транспорт-2013». Ростов н/Д: РГУПС, Т. 2. С. 138.

2. Быкадоров А.Л., Заруцкая Т.А., Муратова-Милехина А.С. Моделирование параметров аварийного режима в тяговой сети переменного тока // Материалы VII Междунар. научно-практ. конф. «Электрификация транспорта «ТРАНСЭЛЕКТРО – 2014». Днепропетровск: ДНУЖ, 2. С. 15.

3. Козлова Ю.С. О необходимости защиты от схлестываний проводов воздушных линий электропередачи // Перспектива развития науки и образования: сборник материалов Международной научно-практической конференции. Душанбе, Ч. 2. С. 56.

4. Ян Дж., Ло Г., Хэ З. Метод классификации неисправностей высоковольтных линий электропередачи на основе вейвлет-энтропии и машины опорных векторов. Power Grid Technol. 2007. С. 23.

5. Ву С., Цао В., Ван Д., Дин М. Метод диагностики неисправностей линий электропередачи на многоопорных векторных или машинных моделях. Высоковольтные технологии. 2020, 46.

6. Способы удаления льда с проводов линий электропередачи: URL: http://ogbus.ru/issues/3_2015/ogbus_3_2015_p794-823_NikitinaIE_ru.pdf.

7. Внедрение автоматизированной системы наблюдения за гололедом в Камышинских электрических сетях. URL: <https://science-education.ru/ru/article/view?id=1246>.

8. Бойнович Л.Б., Емельянов А.М. Методы борьбы с обледенением ЛЭП: перспективы и преимущества новых супергидрофобных покрытий // Электро. 2011. № 6.

9. Левченко И.И. Диагностика, реконструкция и эксплуатация воздушных линий электропередачи в гололедных районах. М.: МЭИ, 2007. 448 с.

10. Чжао Р. Извлечение признаков неисправностей и диагностика линий электропередачи на основе улучшенной модели оптимизации ННО VMD-SVM: магистерская диссертация. Ляский технологический университет, Фусинь, Китай, 2023.

11. Сонг Дж. Метод прогнозирования образования льда и колебаний линий электропередачи на основе метода конечных элементов и улучшенного алгоритма PSO-SVM: магистерская диссертация. Университет «Три ущелья», Ичан, Китай, 2024.

12. Вэй Р., Чен С., Би Г., Дэн С., Ню Ю., Яо Х. Исследование локализации повреждений в линиях сверхвысокого напряжения на основе гибридного глубокого обучения СДКП // Электроэнергетика и электротехника. 2023 . 39 . С. 1–10.

13. Ву Дж., Чжан Ю., Ли С., Ван Ф. Исследование интеллектуального метода диагностики повреждений линий электропередачи на основе СДКП // Крупномасштабные транспортные технологии. 2023. S 2 . С. 62–67.

I.G. Ignatova, M.V. Slyusar

National Research University of Electronic Technology (MIET)

DETECTION AND CLASSIFICATION OF FAULTS IN POWER GRIDS BASED ON THE BAYES–SDCP–SELF-ATTENTION ALGORITHM

Keywords: *power transmission lines, Bayesian optimization, long-term short-term memory networks, attention mechanism, fault detection and classification.*

Abstract. *Power transmission lines, being an essential component of the power grid, play a significant role in ensuring the safe and stable operation of the power grid. To solve the problem of accurately characterizing complex and diverse types of damage, this article proposes a method for detecting and classifying damage to power lines that combines Bayesian reasoning, long-term short-term memory networks, and the attention mechanism. Bayesian reasoning is used to adjust the SDCP hyperparameters, while the SDCP network efficiently processes sequential data using its control mechanism. The self-attention mechanism adaptively assigns weights, focusing on the relationships between information in different sequence positions, capturing global dependencies. The test results show that the proposed Bayes–SDCP–Attention model demonstrates balanced and reliable operation in fault classification.*

НОВЫЕ МЕТОДЫ ДИВЕРСИФИКАЦИИ ПОИСКА В ПОПУЛЯЦИОННЫХ АЛГОРИТМАХ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ. ОБЗОР

Ключевые слова: *глобальная оптимизация, популяционный алгоритм, методы диверсификации поиска, методы интенсификации поиска, методы рандомизации поиска*

Выделяем структурные и параметрические методы диверсификации поиска в популяционных алгоритмах (П-алгоритмах) глобальной оптимизации. Рассматриваем следующие классы структурных диверсифицирующих методов: методы, основанные на программном и/или адаптивном изменении структуры П-алгоритма в процессе эволюции популяции; аналогичные методы, использующие изменение структуры эволюционных операторов; методы на основе синтеза специальных диверсифицирующих эволюционных операторов.

Введение. В настоящее время известно несколько сотен базовых вариантов популяционных алгоритмов (П-алгоритмов), инспирированных живой и неживой природой, а также человеческим обществом. Для каждого из этих алгоритмов предложено большое число модификаций (иногда измеряемое десятками). Обзоры этих алгоритмов представлены, например, в [1, 2].

Исходя из числа популяций, используемых П-алгоритмом, выделяем одно- и многопопуляционные алгоритмы. Все представленные в статье алгоритмы принадлежат классу однопопуляционных.

С точки зрения однородности особей популяции различаем *гомогенные* и *гетерогенные* П-алгоритмы. В гомогенных алгоритмах все особи популяции, за исключением, быть может, их небольшого числа (например, лучшей особи популяции), эволюционируют по одним и тем же правилам (с использованием одних и тех же *эволюционных операторов*). В гетерогенных (*типизированных*) алгоритмах особи различных типов эволюционируют по разным правилам (с использованием разных эволюционных операторов).

Кроме *свободных*, выделяем *управляющие* параметры П-алгоритма, которые могут быть *динамическими*, изменяя свои значения в процессе эволюции популяции *программно, адаптивно* или *программно-адаптивно* [2].

С каждой из сущностей П-алгоритма, исключая его параметры, может быть связана одна или несколько величин – *характеристик*

сущности, значения которых определяют состояние сущности и которые используются для управления эволюцией популяции. Чаще всего используют характеристики популяции и характеристики особей популяции [2].

Используем следующую классификацию П-алгоритмов:

- гомогенные однооператорные алгоритмы, в которых за счет соответствующего изменения значений *управляющих параметров* на начальных итерациях преимущественно обеспечивается диверсификация поиска (исследование области поиска, глобальный поиск), а на завершающих итерациях – интенсификация поиска (эксплуатация, локальный поиск);
- гомогенные двухоператорные алгоритмы, в которых, как правило, первый эволюционный оператор реализует фазу разведки, а второй оператор – фазу эксплуатации;
- гомогенные трех- и более операторные алгоритмы, когда эволюционные операторы в разной степени совмещают функции разведки и эксплуатации;
- гетерогенные алгоритмы, в которых особи одного типа обеспечивают преимущественную разведку, а особи другого типа – эксплуатацию.

Диверсификация поиска в конечном счете призвана повысить вероятность локализации глобального экстремума многоэкстремальной целевой функции. Основными являются *структурные* и *параметрические* методы диверсификации поиска. Структурные методы используют программное и/или адаптивное изменение структуры П-алгоритма и структуры его эволюционных операторов в процессе эволюции популяции. Параметрические методы, аналогично, предполагают изменение значений управляющих параметров. Кроме того, диверсификация поиска может быть обеспечена путем конструирования новых диверсифицирующих эволюционных операторов. Статья посвящена именно указанным средствам диверсификации поиска, используемых в более чем 30 новых (по сравнению с [2]) П-алгоритмах.

Рассматриваем следующие П-алгоритмы.

Гомогенные однооператорные алгоритмы: оптимизационный алгоритм серых волков (Gray Wolf Optimizer, GWO) [3]; роевой оптимизационный алгоритм кенгуру квокка (Quokka Swarm Optimization, QSO) [4]; роевой оптимизационный алгоритм крыс (Rat Swarm Optimizer, RSO) [5]; синус-косинусный алгоритм (Sine Cosine Algorithm, SCA) [6]; алгоритм перетягивания каната (Tug-of-War Optimization, TWO) [7].

Гомогенные двухоператорные алгоритмы: интеллектуальный поисковый алгоритм медуз (Artificial JellyFish Search Algorithm, AJFA)

[8]; оптимизационный алгоритм Архимеда (Archimedes Optimization Algorithm, AOA) [9]; оптимизационный алгоритм бабочек (Butterfly optimization algorithm, BOA) [10]; оптимизационный алгоритм черных вдов (Black Widow Optimization, BWO) [11]; оптимизационный алгоритм жадных людей (Greedy Man Optimization Algorithm, GMOA) [12]; оптимизационный алгоритм Кеплера (Kepler Optimization Algorithm, KOA) [13]; оптимизационный алгоритм пеликанов (Pelican Optimization Algorithm, POA) [14]; оптимизационный алгоритм красноклювых синих сорок (Red-billed Blue Magpie Optimizer, RBMO) [15]; оптимизационный алгоритм стаи песчаных кошек (Sand Cat Swarm Optimization, SCSO) [16]; оптимизационный алгоритм морской звезды (StarFish Optimization Algorithm, SFOA) [17]; алгоритм роя саранчи (SIBL) [18]; алгоритм слизевика (Slime Mould Algorithm, SMA) [19]; сферический поисковый алгоритм (Spherical Search Algorithm, SSA) [20]; оптимизационный алгоритм испарения воды (Water Evaporation Optimization, WEO) [21]; оптимизационный алгоритм китов (Whale Optimization Algorithm, WOA) [22]; оптимизационный алгоритм зебры (Zebra Optimization Algorithm, ZOA) [23].

Гомогенные трех – и более операторные алгоритмы: оптимизационный алгоритм раков (Crayfish Optimization Algorithm, COA) [24]; оптимизационный алгоритм нагуливания жира медведем гризли (Grizzly Bear Fat Increase Optimizer, GBFO) [25]; оптимизационный алгоритм коз (Goat Optimization Algorithm, GOA) [26]; алгоритм морских хищников (Marine Predators Algorithm, MPA) [27]; оптимизационный алгоритм сусликов (Prairie Dog Optimization, PDO) [28]; оптимизационный алгоритм птицы-секретаря (Secretary Bird Optimization Algorithm, SBOA) [29].

Гетерогенные (типизированные) алгоритмы: оптимизационный алгоритм стаи кур (Chicken Swarm Optimization, CSO) [30]; оптимизационный алгоритм навозного жука (Dung Beetle Optimization, DBO) [31]; дифференцированный креативный поиск (Differentiated Creative Search, DCS) [32]; поисковый алгоритм воробьев (Sparrow Search Algorithm, SSA) [33].

1. Постановка задачи. Основные обозначения и соглашения.
Рассматриваем детерминированную задачу безусловной оптимизации

$$\underset{X \in \mathbb{R}^{|X|}}{\text{opt}} f(X) = f(X^*) = f^*,$$

а также аналогичную задачу условной оптимизации

$$\underset{X \in \Pi_X}{\text{opt}} f(X) = f(X^*) = f^*,$$

где $X \in \mathbb{R}^{|X|}$ – вектор варьируемых параметров; $f(X) \in \mathbb{R}^1$ – целевая функция (objective function, cost function); X^*, f^* – искомые

оптимальное решение и соответствующее значение целевой функции;
 $\Pi_X = (X | X^- \leq X \leq X^+) \in \mathfrak{R}^{|X|}$ – параллелепипед допустимых значений вектора X (неравенства понимаются покомпонентно);
 $X^- = (x_1^-, x_2^-, \dots, x_{|X|}^-)$, $X^+ = (x_1^+, x_2^+, \dots, x_{|X|}^+)$ – определенные пользователем $|X|$ -мерные векторы. Полагаем, что фитнес-функция (fitness function) П-алгоритма совпадает с целевой функцией $f(X)$.

Популяцию особей (совокупность кандидатов в решение задачи оптимизации) обозначаем $S = \{s_i, i \in [1:|S|]\}$, где $|S|$ – мощность популяции; s^{gbest} , s^{gworst} – лучшая и худшая особи текущей популяции $S(t)$; $s^{hbest}(t)$, s^{hworst} – аналогичные особи за всю предысторию эволюции популяции S (т.е. на интервале итераций $[0:t]$). Текущее положение особи s_i на итерации t обозначаем $X_i = X_i(t)$, а ее положение на следующей итерации – $\hat{X}_i = X_i(t+1)$; $X^{middle}(t)$ – координаты среднего текущего положения особей популяции.

Используем также следующие обозначения:

t^{max} – заданное максимальное число итераций;

$A \otimes B = (a_1 b_1 \ a_2 b_2 \dots a_{|A|} b_{|B|})$ – прямое произведение векторов A , B одинаковой размерности $|A| = |B|$;

$N_v(m, \sigma)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, распределенных по нормальному закону с математическим ожиданием и СКО, равными m и σ соответственно;

$U_v(a; b)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, равномерно распределенных в интервале $[a; b]$; $b > a$;

$\xi = U_1(0; 1)$ – случайная величина, равномерно распределенная в интервале $[0; 1]$ (полагаем, что всякий раз, когда необходимо значение этой величины, осуществляется ее генерация так что, например, в выражении $a(\xi) + b(\xi)$ значения первой и второй величин ξ , вообще говоря, различны);

$\Xi_v = U_v(0; 1) = (\xi_1, \xi_2, \dots, \xi_v)^T$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, каждое из которых равномерно распределено в интервале $[0; 1]$.

Всюду далее полагаем, что $i \in [1:|S|]$, $j \in [1:|X|]$ (если не оговорено иное).

2. Управление структурой П-алгоритма. Структура П-алгоритма может быть постоянной (стационарной) или изменяться программно, адаптивно или программно-адаптивно по детерминированным и/или стохастическим правилам. Адаптивное изменение структуры

предполагает использование одной или нескольких характеристик популяции [2]. В гомогенных алгоритмах структуру определяет состав используемых эволюционных операторов, а в гетерогенных алгоритмах – текущий набор типов особей и соответствующих им эволюционных операторов. Изменение операторной структуры, очевидно, возможно только в многооператорных гомогенных алгоритмах.

Гомогенные П-алгоритмы. Детерминированная программно-динамическая структура

В алгоритме Архимеда АОА используются операторы «Исследование» и «Эксплуатация». Переключение операторов осуществляем на основе детерминированного программно-динамического фактора переноса (transfer factor) $\varphi(t) = \exp\left(\frac{t-t_{max}}{t_{max}}\right)$:

если имеет место неравенство $\varphi \leq 0,5$, то выполняем первый оператор; в противном случае реализуем второй оператор.

Алгоритм жадных людей GMOA. Оператор «Конкуренция» реализуем на каждой итерации алгоритма, оператор «Мутация» – через каждые Δt итераций (свободный параметр).

Алгоритм испарения воды WEO. В алгоритме используются эволюционные операторы «Однослойная стратегия испарения», «Стратегия на основе капель воды». Переключение операторов производится по номеру текущей итерации: если $t \leq \frac{t_{max}}{2}$, то используется первый оператор, а в противном случае – второй оператор.

Алгоритм морских хищников МРА. Операторы алгоритма МРА соответствуют следующим соотношениям скоростей движения хищника и добычи: «Хищник движется быстрее добычи» (условие реализации $t < \frac{1}{3}t_{max}$); «Хищник и жертва перемещаются с примерно равными скоростями» (условия $\frac{1}{3}t_{max} \leq t < \frac{2}{3}t_{max}$); «Хищник движется медленнее добычи» (условие $t \geq \frac{2}{3}t_{max}$).

Алгоритм сусликов PDO. Операторы «Поиск пищи», «Рытье нор», «Реакция на пищевой сигнал», «Реакция на сигнал о хищниках» реализуем при выполнении условий $t < \frac{t_{max}}{4}$, $\frac{t_{max}}{4} \leq t < \frac{t_{max}}{2}$, $\frac{t_{max}}{2} \leq t < \frac{3t_{max}}{4}$, $\frac{3t_{max}}{4} \leq t \leq t_{max}$ соответственно.

Алгоритм птицы-секретаря SBOA. Оператор «Поиск добычи» выполняем при условии $t < \frac{1}{3}t_{max}$, оператор «Поглощение добычи» – условиями $\frac{1}{3}t_{max} \leq t < \frac{2}{3}t_{max}$, оператор «Нападение на добычу» – условиями $\frac{2}{3}t_{max} \leq t < t_{max}$, оператор «Спасение» – условиями $t = t_{max}$.

Гомогенные П-алгоритмы. Стохастическая программно-динамическая структура

Алгоритм медуз AJFA. Переключение операторов «Движение по океаническому течению», «Движение внутри роя медуз» выполняется с использованием стохастического программно-динамического параметра $c(t) = \text{abs}\left(1 - \frac{t}{t_{\max}}\right)(2\xi - 1)$: при условии $c(t) \geq c_0$ реализуем первый оператор, иначе – второй оператор. Здесь c_0 – свободный параметр алгоритма.

Алгоритм бабочек BOA. На каждой итерации алгоритма оператор «Глобальный поиск» реализуем с вероятностью p (свободный параметр), оператор «Локальный поиск» – в противном случае.

Алгоритм Кеплера KOA. Оператор «Обновление позиций объектов» выполняем при условии $\xi_1 \leq \xi_2$, оператор «Обновление расстояния между объектами и Солнцем» - в противном случае.

Алгоритм стаи песчаных кошек SCSO. Оператор «Поиск добычи» выполняем при условии $\text{abs}(r(t)) > 1$, оператор «Атака на добычу» - в противном случае. Здесь $r(t) = \xi r_g(t)$ – стохастический программно-динамический управляющий параметр; $r_g(t)$ – детерминированный программно-динамический параметр, значения которого убывают с ростом номера итерации t .

Алгоритм морской звезды SFOA. Оператор «Разведка» исполняем при условии $\xi > p$, оператор «Эксплуатация» – в противном случае; p – свободный параметр.

Алгоритм китов WOA. Операторы «Атака» и «Спиральный поиск» переключаются с вероятностью 0,5.

Алгоритм раков COA. Оператор «Отдых» реализуем в условиях $\theta_i > 30$, $\xi < 0,5$; оператор «Соревнование» – в условиях $\theta_i > 30$, $\xi \geq 0,5$; оператор «Питание» – при условии $\theta_i < 30$. Здесь $\theta_i = 15\xi + 20$ – стохастический программно-динамический параметр, которому придается смысл *температуры* (temperature) среды.

Гетерогенные (типизированные) П-алгоритмы

Алгоритм стаи кур CSO. В алгоритме используются особи следующих типов: «Петухи»; «Куры»; «Цыплята». Типы особей обновляем детерминированно и программно после каждых Δt итераций (свободный параметр алгоритма).

Алгоритм навозного жука DBO. Типы особей: «Жук, катящий (навозный) шарик»; «Жук, катящий выводковый шарик»; «Малый навозный жук»; «Вор».

Алгоритм дифференцированного креативного поиска DCS. Типы особей: «Наиболее приспособленные индивиды»; «Индивиды со средней

приспособленностью»; «Наименее приспособленный индивид». Обновление типов индивидов реализуем на каждой итерации алгоритма.

Алгоритм воробьев SSA. Типы воробьев: «Продюсеры» (producers); «Потребители» (scroungers); «Сторожа» (sparrows, which are aware of the danger).

Сферический алгоритм SSA. В алгоритме используются особи двух типов. Особи первого типа эволюционируют в соответствии с оператором «К случайному» (Towards-Rand Search, TRS), особи второго типа – в соответствии с оператором «К наилучшему» (Towards-Best Search, TBS). Оператор TRS использует половину лучших особей популяции, оператор TBS – половину худших особей. Каждый из операторов выполняем на каждой итерации алгоритма.

3. Управление структурой эволюционных операторов. Аналогично структуре П-алгоритма, структура эволюционного оператора может быть стационарной или динамической программной, адаптивной или программно-адаптивной, изменяясь в процессе эволюции популяции по детерминированным и/или стохастическим правилам. Структура операторов алгоритма может изменяться также в зависимости от размерности $|X|$ вектора варьируемых параметров и/или номера особи, как это имеет место, например, в алгоритме морской звезды SFOA. В данном пункте такие ситуации не рассматриваются.

Программно-динамическая структура

Синус-косинусный алгоритм SCA. Эволюционный оператор «Эволюция популяции» (population evolution) определяет итерационная формула

$$\hat{x}_{i,j} = \begin{cases} x_{i,j} + c \sin(2\pi\xi) (2\xi x_j^{gbest} - x_{i,j}), & \text{если } \xi < p, \\ x_{i,j} + c \cos(2\pi\xi) (2\xi x_j^{gbest} - x_{i,j}), & \text{иначе,} \end{cases}$$

где c – детерминированный программно-динамический параметр; p – свободный параметр.

Алгоритм медуз AJFA. Оператор «Движение внутри роя медуз» (movements within a jellyfish swarm):

– если выполняется условие $\xi \geq (1 - c(t))$, то перемещение *пассивно* по формуле

$$\hat{X}_i = X_i + \gamma \xi (X^+ - X^-),$$

где $c(t)$ – стохастический программно-динамический управляющий параметр;

– в противном случае особь перемещается *активно* по формуле

$$\hat{X}_i = X_i + \xi \begin{cases} X_i - X_z, & f(X_i) < f(X_z), \\ X_z - X_p, & \text{иначе.} \end{cases}$$

В последнем выражении $z \in [1: |S|]$ – номер случайной особи текущей популяции; $z \neq i$.

Алгоритм сорок RВМО. Стохастически и программно-динамически изменяют свою структуру операторы «Поиск пищи» и «Нападение на добычу».

Оператор «Поиск пищи» (search for food) имеет два следующих варианта:

– если $\xi < p$, то поиск осуществляют небольшими группами (small groups explore food) из $n \in [2:5]$ особей по формуле

$$\hat{X}_i = X_i + \xi \left(\frac{1}{n} \sum_{j=1}^n X_{i_j} - X_z \right), \quad (1)$$

где p – свободный параметр алгоритма; s_{i_j}, s_z – особи, равномерно случайно выбранные из текущей популяции; $i_j \neq z \neq i$;

-- в противном случае реализует кластерный поиск (cluster search) совокупностью из $m > 10$ особей по формуле

$$\hat{X}_i = X_i + \xi \left(\frac{1}{m} \sum_{j=1}^m X_{i_j} - X_z \right),$$

где обозначения аналогичны обозначениям в (1).

Оператор «Нападение на добычу» (attacking prey) имеет два аналогичных варианта:

– если $\xi < p$, то поиск осуществляем небольшими группами по формуле

$$\hat{X}_i = X^{gbest} + N_1(0, 1) \left(\frac{1}{n} \sum_{j=1}^n X_{i_j} - X_i \right);$$

– в противном случае реализуем кластерный поиск по формуле

$$\hat{X}_i = X^{gbest} + N_1(0, 1) \left(\frac{1}{m} \sum_{j=1}^m X_{i_j} - X_z \right).$$

Алгоритм зебры ZOA. Оператор «Защита от хищников» (defense strategies against predators) используется в двух следующих вариантах:

– если $\xi < p$ (the lion attacks the zebra), то оператор реализует итерационная формула

$$\hat{x}_{i,j} = x_{i,j} + \rho a (2\xi - 1) x_{i,j};$$

– в противном случае (other predators attack the zebra) – формула

$$\hat{x}_{i,j} = x_{i,j} + \xi (x_{z,j} - \text{round}(1 + \xi) x_{i,j}).$$

Здесь ρ – свободный параметр; a – детерминированный программно-динамический управляющий параметр; s_z – случайная особь текущей популяции; $z \neq i$.

Алгоритм морских хищников МРА. Переменную структуру имеет оператор «Частичная «встряска» популяции»:

$$\hat{X}_i = \begin{cases} \hat{X}_i + \gamma(X^- + \Xi_{|x|} \otimes (X^+ - X^-)) \otimes U_{|x|}^{0 \setminus 1}, & \text{если } \xi \leq p, \\ \hat{X}_i + (p(1 - \xi) + \xi)(X_{z_1} - X_{z_2}), & \text{иначе.} \end{cases}$$

Здесь p – свободный параметр алгоритма; s_{z_1}, s_{z_2} – различные случайные особи популяции.

Алгоритм птицы-секретаря SBOA. Переменную структуру имеет оператор «Спасение» (escape strategy):

$$\hat{x}_{i,j} = \begin{cases} x_j^{gbest} + c_3 (2\xi - 1) x_{i,j}, & \text{если } \xi < r, \\ x_{i,j} + \xi (x_{z,j} - \kappa x_{i,j}), & \text{иначе,} \end{cases}$$

где r – свободный параметр; $\kappa = \text{round}(1 + \xi) \in \{1, 2\}$.

Алгоритм воробьев SSA (гетерогенный трехоператорный алгоритм). Программно-динамическую структуру имеют эволюционные операторы, соответствующие следующим типам особей: «Продюсеры» (producers); «Потребители» (scroungers).

Структуру оператора для особей типа «Продюсер» определяет формула

$$\hat{x}_{i,j} = \begin{cases} \exp\left(\frac{-i}{\xi t_{max}}\right) x_{i,j}, & \xi < c_{st}, \\ x_{i,j} + N_1(0, 1), & \text{иначе,} \end{cases}$$

где c_{st} – свободный параметр, имеющий смысл порога безопасности (safety threshold).

Аналогично для особей типа «Потребитель» эволюционный оператор имеет вид

$$\hat{x}_{i,j} = \begin{cases} N_1(0, 1) \exp\left(\frac{x_j^{gworst} - x_{i,j}}{i^2}\right), & i > |S|/2, \\ \hat{x}_j^{pbest} + A(AA^T)^{-1} 1_{|x|}^T (x_{i,j} - \hat{x}_j^{pbest}), & \text{иначе,} \end{cases}$$

где $\hat{X}^{pbest} = X^{pbest}(t + 1)$ – лучшая позиция продюсера; $A = U_{|x|}^{-1 \setminus 1}$.

Алгоритм Кеплера KOA. Оператор «Обновление позиций объектов» (updating objects' positions):

$$\hat{X}_i = X_i + \xi^{\pm 1} \Delta X_i + (g_i + |N_1(0, 1)|) U_{|x|}^{0 \setminus 1} \otimes (X^{gbest} - X_i).$$

Здесь g_i – стохастический программно-адаптивный управляющий параметр; ΔX_i – $|X|$ -мерный вектор скорости особи s_i (velocity), определяемый значением детерминированного адаптивного управляющего параметра $\bar{\rho}_i$, имеющего смысл нормализованного евклидова расстояния $\|X^{gbest} - X_i\|$:

– если $\bar{\rho}_i \leq 0,5$, то

$$\Delta X_i = E_1 \otimes (2\xi X_i - X_{z_2}) + E_2 \otimes (X_{z_1} - X_{z_2}) + \xi^{-1 \setminus 1} (1 - \bar{\rho}_i) U_{|X|}^{0 \setminus 1} \otimes \Xi_{|X|} \otimes (X^+ - X^-); \quad (2)$$

– в противном случае

$$\Delta X_i = \xi \lambda_i (X_{z_1} - X_i) + \xi^{-1 \setminus 1} \xi^{0 \setminus 1} (1 - \bar{\rho}_i) U_{|X|}^{0 \setminus 1} \otimes (\xi X^+ - X^-), \quad (3)$$

где s_{z_1}, s_{z_2} – случайно выбранные особи текущей популяции; $z_1 \neq z_2 \neq i$.

В формулах (2) – (3) приняты следующие обозначения:

$$E_1 = \lambda_i (\xi(1 - \xi) + \xi) U_{|X|}^{0 \setminus 1};$$

$$E_2 = \lambda_i (1_{|X|} - U_{|X|}^{0 \setminus 1}) \otimes (\xi(1_{|X|} - \Xi_{|X|}) + \Xi_{|X|});$$

λ_i – стохастический программно-адаптивный параметр.

Алгоритм пеликанов POA. Оператор «Движение к добыче» (moving towards prey):

$$\hat{x}_{i,j} = \begin{cases} x_{i,j} + \xi (x_{z,j} - \xi^{\pm 1} x_{i,j}), & f(X_z) < f(X_i), \\ x_{i,j} + \xi (x_{i,j} - x_{z,j}), & \text{иначе,} \end{cases}$$

где s_z – случайная особь текущей популяции, общая для всех особей и отличная от особи s_i (жертва (prey)). Изменение структуры оператора осуществляется на основе соответствующего значения целевой функции как характеристики особи.

Алгоритм слизевика SMA. Адаптивно-динамическую структуру имеют эволюционные операторы «Приближение к еде» и «Обволакивание пищи».

Оператор «Приближение к еде» (approach food) реализует итерационная формула

$$\hat{X}_i = \begin{cases} X^{g^{best}} + U_1(-a, a) (w_i X_{z_1} - X_{z_2}), & \xi < p_i, \\ v_c X_i, & \text{иначе.} \end{cases}$$

Здесь $p_i = \tanh(\text{abs}(f(X_i) - f^{h^{best}}))$ – детерминированный адаптивный управляющий параметр; a, v_c – детерминированные программно-динамические параметры; w_i – характеристика особи (масса слизевика), построенная на основе значений целевой функции $f(X)$; s_{z_1}, s_{z_2} – две различные случайные особи текущей популяции; $z_1 \neq z_2 \neq i$. Оператор «Обволакивание пищи» (wrap food):

$$\hat{X}_i = \begin{cases} U_{|x|}(0, 1)(X^+ - X^-) + X^-, & \xi < c_z, \\ X^{gbest} + U_1(-a, a)(w_i X_{z_1} - X_{z_2}), & \xi < p_i, \\ v_c X_i, & \xi \geq p_i. \end{cases}$$

Здесь c_z – свободный параметр алгоритма; первая формула означает, что при реализации данного варианта оператора некоторую часть особей популяции удаляем и вместо удаленных особей создаем такое же число новых особей по правилам инициализации начальной популяции (частичная «встряска» популяции [2]).

Алгоритм раков COA. Оператор «Питание» (foraging) использует два следующих варианта структуры:

– если $q_i > \frac{(c_3+1)}{2}$, то

$$\hat{x}_{i,j} = x_{i,j} + \vartheta_i(\cos(2\pi\xi) - \sin(2\pi\xi))x_j^{hbest},$$

– иначе

$$\hat{x}_{i,j} = \vartheta_i \xi x_{i,j} + \vartheta_i (x_{i,j} - x_j^{hbest}).$$

Здесь $q_i = c_3 \xi \left(\frac{f(x_i)}{f(x^{hbest})} \right)$ – стохастический адаптивный управляющий параметр; c_3 – свободный параметр; ϑ_i – стохастический программно-динамический управляющий параметр.

Алгоритм навозного жука DBO. Эволюционный оператор особей типа «Жук, катящий (навозный) шарик» (ball-rolling dung) имеет переменную структуру, соответствующую режиму качения (rolling) и режиму танцев (dance). При этом последний вариант оператора сам является стохастически программно-динамическим:

– режим качения реализуем с вероятностью 0,9 по формуле

$$\hat{X}_i = X_i + q \xi^{-1} X_i(t-1) + b (X_i - X^{gworst});$$

– режим танцев реализуем в противном случае по формуле

$$\hat{X}_i = \begin{cases} X_i, & \theta \in \{0, \pi/2, \pi\}, \\ X_i + \tan(\theta) (X_i - X_i(t-1)), & \text{иначе.} \end{cases}$$

Здесь b, q – свободные параметры алгоритма; $\theta \in [0; 2\pi]$ – стохастический управляющий параметр, имеющий смысл угла отклонения (deflection angle).

Алгоритм дифференцированного креативного поиска DCS. Переменную структуру имеют эволюционные операторы всех трех типов особей: «Наиболее приспособленные индивиды»; «Индивиды со средней приспособленностью»; «Наименее приспособленный индивид». Структуру оператора «Наиболее приспособленные индивиды» (high-level group) определяет формула

$$\hat{x}_{i,j} = \begin{cases} x_{z,j} + \text{Linnik}(\alpha, \beta), \xi < \eta_i | j = U_1(1: |X|), \\ x_{i,j}, \text{ иначе,} \end{cases}$$

где $\text{Linnik}(\alpha, \beta)$ – распределение Линника с заданными параметрами α, β (свободные параметры алгоритма); η_i – стохастический адаптивно-динамический параметр, построенный на основе такой характеристики особи, как ее *рейтинг* – номер особи в списке особей, отсортированных по убыванию их приспособленности; $z = U_1(1: |S|)$ – номер особи, равномерно случайно выбранной из популяции S и отличной от особей s^{gbest}, s_i .

Аналогично для оператора «Индивиды со средней приспособленностью» (normal group) имеем

$$\hat{x}_{i,j} = \begin{cases} x_j^{gbest} + \lambda (x_{z_1,j} - x_{i,j}) + \xi (x_{z_2,j} - x_{i,j}), \xi < \eta_i | j = U_1(1: |X|), \\ x_{i,j}, \text{ иначе,} \end{cases}$$

где λ – программно-динамический управляющий параметр; z_1 – номер особи, равномерно случайно выбранной из числа особей со средней приспособленностью и отличной от особей s^{gbest}, s_i ; z_2 – номер особи, равномерно случайно выбранной из всей популяции и отличной от особей s^{gbest}, s_{z_1} .

По аналогии для оператора «Наименее приспособленный индивид» (the worst individual) имеем

$$\hat{x}_{|S|,j} = \begin{cases} x_j^- + \xi (x_j^+ - x_j^-), \xi < p_c, \\ x_{|S|,j}, \text{ иначе,} \end{cases}$$

где p_c – свободный параметр алгоритма.

Алгоритм воробьев SSA. Адаптивно-динамическую структуру имеет эволюционный оператор, соответствующий особям типа «Сторожа» (sparrows, which are aware of the danger):

$$\hat{x}_{i,j} = \begin{cases} x_j^{gbest} + N_1(0, 1) (x_{i,j} - x_j^{gbest}), f(X_i) > f(X^{gbest}), \\ x_{i,j} + U_1^{-1} v_{i,j}, f(X_i) = f(X^{gbest}), \end{cases}$$

где $v_{i,j}$ – управляющий адаптивно-динамический параметр.

4. Эволюционные операторы. Почти каждый из рассматриваемых П-алгоритмов использует оригинальные эволюционные операторы. Рассмотрение и анализ каждого из этих операторов не позволяет осуществить размер статьи. С другой стороны, эволюционный оператор в первую очередь определяют его (аддитивные) составляющие: аттракторная; когнитивная; историческая; инерциальная; феромонная; случайная; социальная; когнитивно-социальная; историко-

социальная и другие комбинированные составляющие [2]. Представляем составляющие П-алгоритмов [3-33], которые являются новыми в сравнении с П-алгоритмами, рассмотренными в [2].

Каждая из составляющих П-алгоритма определяет, во-первых, некоторое направление перемещения особи, а во-вторых – величину шага ее перемещения в этом направлении, который, по общим правилам, может быть детерминированным или стохастическим, стационарным, программно-динамическим, адаптивно-динамическим и программно-адаптивным. Ограничиваемся рассмотрением только направления перемещения особи, которое определяет данная составляющая.

Случайная составляющая

Алгоритм перетягивания каната TWO (оператор «Конкуренция и вытеснение»):

$$N_{|x|}(0, 1) \otimes (X^+ - X^-).$$

Аналогичные составляющие используются в *алгоритме медуз* (оператор «Движение внутри роя медуз»), *алгоритме Кеплера* КОА (оператор «Обновление позиций объектов»), *алгоритме слизевика* SMA (оператор «Обволакивание пищи»), *алгоритме коз* GOA (оператор «Адаптивная кормовая стратегия»), *алгоритме морских хищников* МРА (оператор «Частичная «встряска» популяции»).

Социальная составляющая

Алгоритм волков GWO (оператор «Охота»):

$$2\Xi_{|x|} \otimes X_{\mu} - X_i,$$

где X_{μ} – положение одной из трех наиболее успешных особей текущей популяции; $\mu \neq i$.

Алгоритм медуз AJFA (оператор «Движение по океаническому течению»). В алгоритме используются две следующие оригинальные социальные составляющие:

$$X^{gbest} - \beta \xi X^{middle}; \quad \xi \begin{cases} X_i - X_z, & f(X_i) < f(X_z), \\ X_z - X_i, & \text{иначе.} \end{cases}$$

Здесь β – свободный параметр алгоритма; $z \in [1: |S|]$ – номер случайной особи текущей популяции; $z \neq i$.

Алгоритм бабочек BOA (оператор «Локальный поиск»):

$$\xi^2 X_{z_1} - X_{z_2},$$

где X_{z_1}, X_{z_2} – текущие позиции двух случайно выбранных особей популяции, отличных от особи s_i . Аналогичные составляющие использованы в *алгоритме Кеплера* КОА (оператор «Обновление позиций объектов»), *алгоритме слизевика* SMA (операторы

«Приближение к еде», «Обволакивание пищи») и ряде других алгоритмов.

В алгоритме Кеплера КОА (оператор «Обновление расстояния между объектами и Солнцем») находят применение две следующие социальные составляющие:

$$\frac{X_i + X^{gbest} + X_{z_1}}{3}, \quad \frac{X_i + X^{gbest} + X_{z_1}}{3} - X_{z_2}.$$

Здесь X_{z_1}, X_{z_2} – текущие позиции двух случайно выбранных особей популяции; $z_1 \neq z_2 \neq i$ /

В алгоритме сорок RBMO (операторы «Поиск пищи», «Нападение на добычу») применяются близкие социальные составляющие

$$\xi \left(\frac{1}{n} \sum_{j=1}^n X_{i_j} - X_z \right); \quad N_1(0, 1) \left(\frac{1}{n} \sum_{j=1}^n X_{i_j} - X_i \right),$$

где $n > 1$ – целый свободный параметр алгоритма; s_{i_j}, s_z – особи, равномерно случайно выбранные из текущей популяции; $i_{j1} \neq z \neq i$ /

Алгоритм испарения воды WEO (операторы «Однослойная стратегия испарения», «Стратегия на основе капель воды»):

$$\xi \left(\mathbf{X}[U_1(1:|S|), U_1(1:|X|)] - \mathbf{X}[U_1(1:|S|), U_1(1:|X|)] \right).$$

Здесь $\mathbf{X} = (X_1 \ X_2 \ \dots \ X_{|S|})^T$ – матрица популяции размерности $|S| \times |X|$.

Алгоритм зебры ZOA. В операторах «Поиск пищи», «Защита от хищников» этого алгоритма использованы социальные составляющие вида

$$\xi \left(x_j^{gbest} - \text{round}(1 + \xi) x_{i,j} \right), \quad \xi \left(x_{z,j} - \text{round}(1 + \xi) x_{i,j} \right)$$

соответственно, где s_z – случайная особь текущей популяции, отличная от особи s_i .

Алгоритм раков COA (оператор «Отдых»):

$$\xi \left(x_{i,j}^{shade} - x_{i,j} \right),$$

где $X_i^{shade} = \frac{1}{2} (X^{hbest} - X^{gbest})$ – координаты средней точки между указанными позициями особей s^{hbest}, s^{gbest} .

Алгоритм медведя гризли GBFO. Оригинальные социальные составляющие содержат операторы «Поиск местонахождения овощей, фруктов, моллюсков, прудов, рек для ловли рыбы и отслеживание перемещения рыбы», «Охота и забота о потомстве».

Оператор «Поиск местонахождения овощей, фруктов, моллюсков, прудов, рек для ловли рыбы и отслеживание перемещения рыбы:

$$\sum_{j=1}^5 r_j (X_{i_j} - X_i),$$

где $s_{i,j}, j \in [1:5]$ – пять лучших особей текущей популяции; $r_j, j \in [1:5]$ – стохастические весовые коэффициенты.

Оператор «Охота и защита детенышей»:

$$\Delta X_i^1 = 2\Xi_{|X|} \otimes X_{i_1} - \Xi_{|X|} \otimes X_{i_2},$$

где s_{z_1}, s_{z_2} – особи, равномерно случайно выбранные из популяции S , и s_{z_2} – лучшая из этих особей; $z_1 \neq z_2 \neq i$.

Алгоритм коз GOA (оператор «Прыжки»):

$$\gamma(X_z - X_i),$$

где s_z – особь, равномерно случайно выбранная из текущей популяции и отличная от особи s_i ; γ – свободный детерминированный параметр.

Алгоритм морских хищников МРА. Элементами новизны обладают социальные составляющие следующих эволюционных операторов данного алгоритма.

«Хищник движется быстрее добычи»:

$$X^{gbest} - N_{|X|}(0, 1) \otimes X_i.$$

«Хищник движется примерно с той же скоростью, что и добыча»:

$$X^{gbest} - Levy_{|X|}(\alpha) \otimes X_i, ; i \in \left[1: \frac{|S|}{2}\right];$$

$$N_{|X|}(0, 1) \otimes X^{gbest} - X_i, i \in \left[\frac{|S|}{2} + 1: |S|\right].$$

Здесь $Levy_{|X|}(\alpha)$ – $|X|$ -мерное распределение Леви со свободным параметром α .

«Хищник движется медленнее добычи»:

$$Levi_{|X|}(\alpha) \otimes X^{gbest} - X_i.$$

Аналогичную составляющую использует алгоритм сусликов PDO (оператор «Рытье нор»).

Алгоритм сусликов PDO (операторы «Поиск пищи», «Реакция на пищевой сигнал»):

$$x_j^{gbest} + \frac{x_{i,j} x_j^{mean}}{x_j^{gbest} (x_j^+ - x_j^-) + \delta}, \quad \frac{x_j^{gbest} - x_{z,j}}{x_j^{gbest} + \delta},$$

где δ – свободный параметр алгоритма; s_z – случайная особь текущей популяции, отличная от рассматриваемой особи.

Алгоритм птицы-секретаря SBOA (операторы «Нападение на добычу», «Спасение»):

$$c_2 Levy x_{i,j} + x_j^{gbest}, \quad \xi (x_{z,j} - \kappa x_{i,j}),$$

где c_2 – детерминированный программно-динамический управляющий параметр; $\kappa = round(1 + \xi) \in \{1, 2\}$.

Алгоритм навозного жука DBO. Признаками новизны обладают социальные составляющие следующих операторов.

Оператор «Жук, катящий выводковый шарик»:

$$X_i - \max\left((1-r)X^{gbest}, X^-\right); \quad X_i - \min\left((1+r)X^{gbest}, X^+\right),$$

где r – детерминированный программно-динамический управляющий параметр, линейно убывающий с ростом t .

Оператор «Малый навозный жук»:

$$X_i - \max\left((1-r)X^{hbest}, X^-\right); \quad X_i - \min\left((1+r)X^{hbest}, X^+\right).$$

Сферический алгоритм SSA. Оператор «К случайному»:

$$P_i(X_{z_1} + X_{z_2} - X_{z_3} - X_i),$$

где $s_{z_1}, s_{z_2}, s_{z_3}$ – случайные особи текущей популяции; $P_i - (|X| \times |X|)$ – проекционная матрица [31]; $z_1 \neq z_2 \neq z_3 \neq i$.

Оператор «К наилучшему»:

$$P_i(X_{z_1}^{best} + X_{z_2} - X_{z_3} - X_i),$$

где $s_{z_1}^{best}$ – случайная особь из числа n_p (свободный параметр) лучших особей текущей популяции; $z_1 \neq i$.

Заключение. Кроме рассмотренных для диверсификации поиска в П-алгоритмах могут использоваться следующие средства.

Рандомизация эволюционных операторов. В П-алгоритмах обычно используют рандомизацию текущего положения особи i , быть может, скорости и ускорения. Различают методы аддитивной, мультипликативной и смешанной рандомизации переменных состояния особей популяции [2]. Чаще всего в процедуре рандомизации используют равномерно распределенные случайные величины, но широко применяют также нормально распределенные величины. Находят применение и иные законы распределения, например, распределение Леви. Наряду со случайными последовательностями находят применение хаотичные последовательности.

Управляющие переменные. Для обеспечения баланса диверсификационных и интенсификационных свойств в П-алгоритмах широко используют программные, адаптивные и программно-адаптивные управляющие переменные, которые могут быть детерминированными, стохастическими или построенными на основе той или иной хаотической последовательности.

Характеристики особи и популяции. Эффективность П-алгоритма, в том числе соотношение его диверсификационных и интенсификационных свойств, может в значительной степени зависеть от характеристик особей и популяции, используемых его эволюционными операторами.

Размеры статьи не позволили рассмотреть эти средства П-алгоритмов.

Литература

1. Xing B., Gao W.-J. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer Cham Heidelberg New York Dordrecht London, 2014. 451 p.
2. Карпенко А.П.. Анализ и синтез популяционных алгоритмов глобальной оптимизации. В 2-х т. М.: Издательство МГТУ им. Н.Э. Баумана, 2024. Т. 1. 368 с., Т. 2. 387 с.
3. Mirjalili S., Mirjalili S.M., Lewis A. Grey Wolf Optimizer // Advances in Engineering Software, 2014. Vol. 69. P. 46 – 61.
4. Wijdan Jaber AL-kubaisy, Belal AL-Khateeb. Quokka swarm optimization: A new nature-inspired metaheuristic optimization algorithm // Journal of Intelligent Systems, 2024. - 33: 20240051. URL: <https://doi.org/10.1515/jisys-2024-0051>.
5. Dhiman G., Garg M., Nagar A., Kumar V., Dehghani M.. A novel algorithm for global optimization: Rat Swarm Optimizer // Intell Human Comput, 2020. Vol. 12. P. 8457-8459.
6. Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems // Knowl-Based Syst., 2016. 96:120–133. URL: [https://doi.org/10.1016/j.knsys.\(2015.12.022\)](https://doi.org/10.1016/j.knsys.(2015.12.022)).
7. Kaveh A., Zolghadr A. Novel meta-heuristic algorithm. Tug of war optimization: a new metaheuristic algorithm // International Journal of Optimization in Civil Engineering, 2016. Vol. 6(4). P. 492.
8. Chou J.S, Truong D.N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean // Appl Math Comput., 2021. 389:125535.
9. Fatma A. Hashim, Kashif Hussain, Essam H. Houssein. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems // Springer Link, 2020. P. 23-34.
10. S. Arora, S. Singh. Butterfly optimization algorithm: a novel approach for global optimization // Soft Computing, 2019. Vol. 23. P. 715–734. URL: <https://doi.org/10.1007/s00500-018-3102-4>.
11. Hayyolalam V., Pourhaji Kazem A.A. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems // Engineering Applications of Artificial Intelligence, 2020. Vol. 87. Article 103249.
12. Hammed Nozari, Hossein Abdi. Greedy Man Optimization Algorithm (GMOA): A Novel Approach to Problem Solving with Resistant Parasites // Journal of Industrial and Systems Engineering, 2024. Vol. 16, No. 3, P. 106-117.
13. Abdel-Basset M., Reda Mohameda, Shaimaa A. Abdel Azeem, Mohammed Jameel, Mohamed Abouhawwash. Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion // Knowledge-Based Systems, 2023. Vol. 268. Article 110454.
14. Trojovský P., Dehghani M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications // Sensors, 2022. Vol. 22. Article 855.
15. Fu S., Li K., Huang H., Ma C., Fan Q., Zhu Y. Red-billed blue magpie optimizer: a novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems // Artificial Intelligence Review, 2024. Vol.57. P. 1–89.
16. Seyyedabbasi A., Kiani F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems // Engineering with Computers, 2020. P. 1–25. URL: <https://doi.org/10.1007/s00366-022-01604-x>.

17. SFOA Changting Zhong, Gang Li, Zeng Meng, Haijiang Li, Ali Riza Yildiz, Seyedali Mirjalili. Starfish Optimization Algorithm (SFOA): A Bio-Inspired Metaheuristic Algorithm for Global Optimization Compared with 100 Optimizers // *Neural Computing and Applications*, 2024. Vol. 37. P. 3641–3683.
18. Курейчик В.В., Родзин С.И. Модель коллаборативного поведения роя саранчи для оптимизации многомерных мультиэкстремальных функций // *Изв. вузов. Сев.-Кавк. регион. Техн. науки*. 2023. № 1. С. 10–16. URL: <http://dx.doi.org/10.17213/1560-3644-2023-1-10-16>.
19. Shimin Li¹, Huiling Chen¹, Mingjing Wang, Ali Asghar Heidari, Seyedali Mirjalili. Slime mould algorithm: A new method for stochastic optimization, *Future Generation Computer Systems*, 2020. URL: <https://doi.org/10.1016/j.future.2020.03.055>.
20. Kumar A., Misra R.K., Singh D., Mishra S., Das S. The spherical search algorithm for bound-constrained global optimization problems // *Appl. Soft Comput.*, 2019. Vol. 85 105734.
21. Kaveh A., Bakhshpoori T. Water Evaporation Optimization: A novel physically inspired optimization algorithm / *Computers & Structures*, 2016. P. 69-85.
22. Mirjalili S., Lewis A. The Whale Optimization Algorithm // *Adv. Eng. Softw.*, 2016. Vol. 95. P. 51-67. URL: <https://doi.org/10.1016/j.adveng-601 soft.2016.01.008>.
23. Trojovská E., Dehghani M., Trojovský P. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm // *IEEE Access*, 2022, Vol. 10: 49445-49473.
24. Heming Jia, Honghua Rao, Changsheng Wen, Seyedali Mirjalili. Crayfish optimization algorithm // *Artificial Intelligence Review*, 2023. Vol. 56. P. 1919–1979. <https://doi.org/10.1007/s10462-023-10567-4>.
25. Moslem Dehghani, Mokhtar Aly, Jose Rodriguez, Ehsan Sheybani, Giti Javidi. A Novel Nature-Inspired Optimization Algorithm: Grizzly Bear Fat Increase Optimizer // *Biomimetics*, 2025. Vol. 10(6). Article 379. <https://doi.org/10.3390/biomimetics10060379>.
26. Hamed Nozaria, Hoessein Abdib, Agnieszka Szmelter-Jarosz. Goat Optimization Algorithm: A Novel Bio-Inspired Metaheuristic for Global Optimization // *Applied Innovations in Industrial Management*, 2025. Vol. 5, No. 1, pp. 70-80.
27. Faramarzi A., Heidarinejad M., Mirjalili S., Gandomi A.H. Marine Predators Algorithm: A nature-inspired Metaheuristic, *Expert Systems with Applications*, 2020. Vol. 152. Article No. 113377. URL: <https://doi.org/10.1016/j.eswa.2020.113377>.
28. Ezugwu A.E., Agushaka J.O., Abualigah L., Mirjalili S., Gandomi A.H. Prairie dog optimization algorithm // *Neural Computation*, 2022. Vol. 34 (22). P. 20017–20065.
29. Fu Y., Liu D., Chen J., He L. Secretary Bird Optimization Algorithm: a new metaheuristic for solving global optimization problems // *Artificial Intelligence Review*, 2024. Vol. 57. P.123-137. URL: <https://doi.org/10.1007/s10462-024-10729-y>
30. Meng X.; Liu Y., Gao X., Zhang H. A New Bio-Inspired Algorithm: Chicken Swarm Optimization // *Lecture Notes in Computer Science*, 2014. Vol. 8794. P. 86–94. URL: <https://www.researchgate.net/publication/278691165>
31. Xue J., Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization // *The Journal of Supercomputing*, 2023. Vol. 79. P. 7305-7336. URL: <https://doi.org/10.1007/s11227-022-04959-6>.
32. Duankhan P., Suna, K., Chiewchanwattana S., Nasa-ngium P. The Differentiated Creative Search (DCS): Leveraging Differentiated Knowledge-Acquisition and Creative Realism to Address Complex Optimization Problems // *Expert Systems with Applications*, 2024. Vol. 252. Article 123734. doi: 10.1016/j.eswa.2024.123734.

33. Xue J. & Shen B. A novel swarm intelligence optimization approach: Sparrow search algorithm. Syst. Sci. Control Eng., 2020. Vol. 8(1), P. 22–34. URL: <https://doi.org/10.1080/21642583.2019.1708830>.

A.P. Karpenko
Bauman Moscow State Technicaly

ON THE PROBLEM OF CARLEMAN TYPE FOR QUASIHARMONIC FUNCTIONS OF THE GENUS n

Keywords: *global optimization, population algorithm, search diversification methods, search intensification methods, search randomization methods.*

Abstract. *We distinguish structural and parametric methods for diversifying search in population-based algorithms (P-algorithms) of global optimization. We consider the following classes of structural methods: methods based on programmatic and/or adaptive modification of the P-algorithm structure during population evolution; similar methods that utilize changes in the structure of evolutionary operators; and methods based on the synthesis of special diversifying evolutionary operators.*

А.К. Киселева, В.И. Мунерман
Смоленский государственный университет

УДК 510.522

АЛГОРИТМИЧЕСКАЯ ТЕОРИЯ СЛУЧАЙНОСТИ В ПРАКТИЧЕСКОМ АНАЛИЗЕ ДАННЫХ

Ключевые слова: *анализ данных, статистический анализ, алгоритмическая случайность, колмогоровская сложность, мартин-лефовский тест.*

В статье рассматриваются теоретические и практические подходы к анализу данных стационарных эргодических источников (генераторы случайных и псевдослучайных чисел, марковские процессы, прикладные источники), в результате чего предлагается математическая модель метода анализа источников данных на основе алгоритмической теории случайности.

Введение. Понятие случайности является фундаментальным не только для теории вероятностей и математической статистики, но и для математической и общей философии, а также для прикладных областей,

связанных с анализом данных, криптографией или машинным обучением. В классической теории вероятностей термин «случайность» обозначает изначальное свойство модели, а вероятностные утверждения формулируются для всех её реализаций. Однако для анализа конкретных объектов, например, битовых последовательностей, полученных в результате измерений, симуляций или работы алгоритмов, такой подход оказывается недостаточным. Чтобы найти решение данной проблемы, необходимо обратиться к другим областям физико-математических наук.

Понятие энтропии из теории информации позволяет утверждать, что истинно случайный объект – наиболее информационно емкий. Таким образом, если последовательность можно описать короче (сжать), это означает, что в ней меньше информации, чем в «истинно случайной». На этой идее основано понятие колмогоровской сложности [3, 10], согласно которому конечный объект считается случайным, если его кратчайшее описание не короче самого объекта (с точностью до аддитивной константы). Связь между энтропией и алгоритмической сложностью формализована, в частности, в работах Брудно [6].

Интуитивно можно представить следующую конструкцию: если есть истинно случайный генератор битов, то вывести результат его работы с помощью алгоритма можно только буквально, получая каждый бит из генератора, то есть, получив программу той же длины, что и результат работы генератора. Данный подход продолжается в теории алгоритмической случайности, предложенной П. Мартином-Лефом. В частности, мартин-лефовские тесты или *ML*-тесты – система построения уровней проверки, которая в предельном случае выявляет любое нетипичное для случайной поведение последовательности [1, 2].

При таком подходе к определению случайности можно также контролировать и уровень ошибки. К примеру, согласно теореме Рябко [4, 5] разница между длиной строки и длиной порождающего ее универсального кода удовлетворяет следующему условию:

$$\forall \alpha P_{H_0} \{n - |\phi(x_n)| \geq -\log_2 \alpha\} \leq \alpha,$$

где α – уровень значимости. Это фактически дает готовую формулу, связывающую возможность сжатия данных и *p-value* (вероятность получить наблюдаемый результат при условии верности нулевой гипотезы), но, что более важно, позволяет утверждать, что использование универсального кода, например современных архиваторов (вроде *Lempel-Ziv*) [7] или контекстного моделирования (*PPM*, *LZMA*), валидно для оценки алгоритмической сложности.

Постановка задачи. Классическим результатом теории алгоритмической случайности является эквивалентность колмогоровской сложности и универсального *ML*-теста в вопросе подхода к определению случайности. Но даже приводя полный

теоретический критерий, они не находят практического применения, так как в общем случае оба не вычислимы (в силу неразрешимости известной проблемы остановки).

На практике чаще всего используются тесты математической статистики. Стоит отметить, что любой статистический тест рассматривается как частный случай *ML*-теста, который фиксирует конкретную вычислимо обнаружимую закономерность. Такой тест, который согласуется с определением, задает эффективно перечислимое множество «плохих» префиксов, которое обладает малой вероятностной мерой, и тем самым исключает последовательности, демонстрирующие соответствующее нетипичное для случайности поведение.

Самое распространенное практическое решение задачи определения случайности последовательности – это статистическая батарея *NIST STS*. Она включает в себя 15 тестов и де-факто считается стандартом проверки генераторов псевдослучайных чисел. Тем не менее она проверяет только определенные статистические свойства, в результате чего зачастую требуется дополнительный анализ, а также работает лишь со строками длиной более 10^6 битов. Универсальные методы сжатия обладают более общим характером: они выявляют закономерности, проявляющиеся в избыточности описания, что согласуется с принципами универсального кодирования [7, 9].

Формализация модели. В данной работе вместо универсального, но невычислимого *ML*-теста или фиксированной батареи предлагается строить систему из уровней проверок, объединяя статистический и алгоритмический подходы. Ожидается, что при достаточной длине последовательности подобная система выявит те же классы нетипичных последовательностей, что и классические алгоритмические критерии случайности, в частности *ML*-тесты и критерий колмогоровской сложности, а также сработает не хуже популярных статистических батарей.

Для каждого уровня модели задается минимально необходимая длина последовательности, при которой статистическая мощность уровня становится достаточной для выявления закономерностей данного типа. Проверка итеративна: на каждом уровне по теореме Рябко вычисляется *p-value*, и, если оно меньше изначально заданного уровня значимости, последовательность признается нетипичной (неслучайной), а вычисления прекращаются. Если же *p-value* больше уровня значимости и длина достаточна, управление передается на следующий уровень. Если длина недостаточна для следующего уровня, процедура завершается выводом о локальной типичности данных на пройденном уровне.

Уровни формализуются следующим образом:

1) L_1 использует простейшие модели с нулевым контекстом для выявления частотных дисбалансов символов;

2) L_2 базируется на алгоритмах семейства *Lempel-Ziv* (например, *LZ77*), целью которого является поиск коротких повторяющихся паттернов и структур самоподобия. Для исключения влияния системных заголовков архиватора длина его работы без заголовка вычисляется как разность между сжатым массивом и сжатой пустой строкой;

3) L_3 задействует мощные методы аппроксимации (*LZMA*, *PPM*), способные учитывать марковские зависимости высоких порядков на больших дистанциях;

4) L_4 с помощью *PAQ* (семейство алгоритмов сжатия данных с открытым исходным кодом) использует несколько независимых моделей (модели разреженных контекстов, нейросети) и смешивает их прогнозы через логистическую регрессию или адаптивные веса. На этом уровне библиотека способна выявлять структурную неслучайность, недоступную классическим статистическим тестам [8].

Таким образом, L_1 выявляет простейшие статистические закономерности на данных даже малой длины, L_2 работает на уровне простейших архиваторов и идентифицирует закономерности со сдвигами. По мере перехода к уровням L_3 и L_4 модель стремится к аппроксимации колмогоровской сложности в рамках выбранного класса моделей, расширяя класс выявляемых зависимостей по сравнению с фиксированными статистическими тестами. Кроме того, алгоритм не просто констатирует случайность или неслучайность данных, а оценивает информационную емкость последовательности, что дает их более полную оценку.

Программные методы. Для практической реализации удобно представить систему в библиотеке языка *Python*, являющегося популярным для решения задач в области *data science*, а также имеющего развитую экосистему математических пакетов и возможности нативной интеграции с библиотеками сжатия. Программная архитектура в таком случае строится на принципе адаптивного конвейера (*pipeline*): каждый уровень инкапсулирован в отдельный модуль, реализующий интерфейс его математического аппарата. Уровень L_1 кодируется на чистом *Python* с использованием встроенной библиотеки *math*, для уровней $L_2 - L_3$ используются оптимизированные C-расширения стандартной библиотеки (*zlib*, *lzma*), а уровень L_4 реализуется через абстракцию внешних вызовов высокоэффективного движка *ZPAQ*. При этом допускаются более эффективные варианты реализации, что дает пространство для программного совершенствования модели.

Заключение. Таким образом, рассмотрение практической проблемы исследования конечных последовательностей на случайность в рамках алгоритмической теории информации позволяет предложить новые методы тестирования данных на основе универсального кодирования и реализовывать их с использованием современных

архиваторов. Такие методы имеют единое теоретическое основание в рамках алгоритмической случайности, дают возможность контроля уровня ошибки через длину универсального кода, позволяют полноценно оценивать уровень энтропии источника и допускают эффективную программную реализацию с использованием внешних библиотек.

Литература

1. Верецагин Н.К., Успенский В.А., Шень М. Колмогоровская сложность и алгоритмическая случайность. М.: МЦНМО, 2013. 576 с.
2. Успенский В.А. Четыре алгоритмических лица случайности. Дубна: Летняя школа «Современная математика». 2005. 48 с.
3. Li Ming, Vitányi, Paul M.B. An introduction to Kolmogorov complexity and its applications. New York: Springer-Verlag, 1997. 654 p.
4. Ryabko B.Ya. Prediction of random sequences and tests for randomness. Problems of Information Transmission, 1988. P. 3–14.
5. Рябко Б.Я., Фионов А.С. Основы современной криптографии и стеганографии. М.: 2013, Горячая линия-Телеком.
6. Brudno A.A. Entropy and the complexity of the trajectories of a dynamical system, Transactions of the Moscow Mathematical Society, 1983. P. 127–151.
7. Ziv J., & Lempel A. A universal algorithm for sequential data compression. IEEE Transactions on Information Theory, 1977. P. 337–343.
8. Cilibrasi R., & Vitányi P. Clustering by compression. IEEE Transactions on Information Theory, 2005. P. 1523–1545.
9. Cleary J., & Witten I. Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications, 1984. P. 396–402.
10. Kolmogorov A.N. Logical basis for information theory and probability theory // IEEE Transactions on Information Theory, 1968. 14. P. 662– 664.

A.K. Kiseleva, V.I. Munerman
Smolensk State University

ALGORITHMIC THEORY OF RANDOMNESS IN PRACTICAL DATA ANALYSIS

Keywords: *data analysis, statistical analysis, algorithmic randomness, Kolmogorov complexity, martin-lof test.*

Abstract. *The article examines practical and theoretical approaches to the analysis of data from stationary ergodic sources (random and pseudo-random number generators, Markov processes, applied sources), as a result of which a mathematical model of the method of analyzing data sources based on the algorithmic theory of randomness is proposed.*

**РАЗРАБОТКА МЕТОДА БАЛАНСИРОВКИ НАГРУЗКИ
В ОБЛАЧНЫХ СИСТЕМАХ НА ОСНОВЕ
МОДИФИЦИРОВАННОГО АЛГОРИТМА
FALCON OPTIMIZATION**

Ключевые слова: *балансировка нагрузки, облачные вычисления, сверточные нейронные сети, алгоритм, обучение с подкреплением, виртуальные машины.*

Облачные вычисления являются масштабируемой и экономичной платформой, которая предоставляет доступ к общим вычислительным ресурсам в соответствии с запросом пользователя. Эффективная балансировка нагрузки необходима для реализации оптимальной производительности и максимального использования ресурсов. Она обеспечивает равномерное распределение сетевого трафика между серверами, предотвращает перегрузку, увеличивает время отклика и повышает надежность системы. В статье предлагается метод балансировки нагрузки в облачных системах на основе модифицированного алгоритма Falcon Optimization и обучения с подкреплением. Предложенная модель использует сверточные и рекуррентные нейронные сети для прогнозирования нагрузки на виртуальные машины и динамической классификации их по категориям перегруженности и недогрузки. Предлагаемый подход значительно улучшает планирование задач и балансировку нагрузки по сравнению с существующими методами, такими как алгоритм оптимизации нагрузки (LOA), подкрепление с обучением (RL) и алгоритм оптимизации Falcon (FOA).

Комплексные задачи по расстановке приоритетов, оптимизации использования памяти, минимизации времени выполнения и балансировке загрузки ЦП значительно повышают эффективность планирования задач в средах облачных вычислений. Благодаря метаобучению с подкреплением эти задачи обеспечивают эффективное распределение ресурсов, сокращают время выполнения других задач и поддерживают сбалансированную нагрузку в динамичных и разнородных облачных системах. Метаобучение с подкреплением позволяет адаптивно принимать решения, благодаря чему система может динамически корректировать стратегии планирования с учетом изменений рабочей нагрузки в реальном времени и меняющихся ограничений ресурсов [1]. Такая адаптивность повышает общую

эффективность использования ресурсов и надёжность системы, а также помогает справиться с проблемами, возникающими из-за нестабильной рабочей нагрузки и непредсказуемых требований к задачам. Однако эти цели предполагают компромиссные решения. Например, приоритизация критически важных задач может увеличить продолжительность проекта или привести к нехватке ресурсов. По мере роста сложности системы эти компромиссы становятся всё более очевидными, что требует применения методов оптимизации. Предложенный подход объединяет метаобучение с подкреплением и гибридную оптимизацию для динамической адаптации к изменениям рабочей нагрузки, оптимизации параметров качества обслуживания и минимизации неэффективности и частоты сбоев. Эта гибридная методология обеспечивает принятие оптимальных решений в сложных облачных средах, повышая масштабируемость, скорость отклика и общую производительность системы.

Для поддержания оптимальной производительности системы и эффективного использования ресурсов в облачных вычислениях необходимо решить ряд проблем. Одна из ключевых проблем – разнообразие облачных центров обработки данных, в которых ресурсы обладают разной мощностью и возможностями. Чтобы избежать перегрузки одних ресурсов и недостаточного использования других, необходимо эффективно распределять задачи между этими разнообразными ресурсами. Управление изменяющимися рабочими нагрузками в облачной среде представляет серьёзную проблему.

Требования к рабочей нагрузке могут меняться со временем, поэтому необходимы методы балансировки нагрузки, способные динамически адаптировать распределение ресурсов к меняющимся требованиям. Такая адаптивность крайне важна для предотвращения снижения производительности в периоды пиковой нагрузки. Масштабируемость – ещё одна проблема, поскольку сервисы облачных вычислений должны обслуживать растущее число клиентов и приложений. Для удовлетворения растущего спроса и эффективного распределения задач в расширяющейся инфраструктуре необходимы масштабируемые решения для балансировки нагрузки.

Достижение эффективной балансировки нагрузки с минимальными накладными расходами имеет важное значение для реализации высокопроизводительных сред облачных вычислений. Чтобы сделать формулировку проблемы более понятной, важно упомянуть о проблеме, связанной с качеством обслуживания (QoS), например в стохастическом планировщике с учётом энергопотребления для пакетов заданий с ограничениями по приоритету в гетерогенной вычислительной системе, а также в энергоэффективных алгоритмах планирования для пакетных приложений (VoT) в гетерогенных вычислительных системах. Это

позволит более чётко и конкретно сформулировать стоящие перед нами задачи.

В исследовании используется подход динамической кластеризации для повышения эффективности кластеризации и скорости реагирования на изменение рабочей нагрузки. Кроме того, использование гибридного алгоритма позволяет решить проблемы оптимизации.

Целевая функция формулируется с использованием таких показателей качества обслуживания, как время отклика, пропускная способность и доступность. Время отклика – это промежуток времени, в течение которого сервис отвечает на запрос определённого типа. Время отклика определяется интенсивностью нагрузки, которая может измеряться количеством одновременных запросов или частотой запросов (количество запросов в секунду). Качество обслуживания учитывает не только среднее время отклика, но и процентиль времени отклика.

$$T_{vgRT} = \frac{T_{RT}}{N}, \quad (1)$$

где T_{vgRT} – это среднее время отклика, T_{RT} - общее время отклика, а N – общее количество запросов;

Пропускная способность – это максимальная скорость обработки данных сервисом. Показатели качества обслуживания, которые часто включают временной интервал (T), используются для анализа того, как меняется пропускная способность в зависимости от интенсивности нагрузки и пиковой пропускной способности системы

$$W_{RT} = \frac{N}{T}, \quad (2)$$

где T – это общий временной интервал;

Доступность системы или сервиса — это процентное соотношение времени, в течение которого система или сервис работают и доступны для пользователей. Обычно это выражается в процентах времени безотказной работы. Высокая доступность крайне важна для критически важных сервисов, поскольку она сводит к минимуму время простоя и перерывы в обслуживании, обеспечивая при этом безупречную работу пользователей.

$$A_{RT} = \frac{T_{Ut}}{T_{Ut} + T_{Dt}} \cdot 100\%, \quad (3)$$

где T_{Ut} – это общее время безотказной работы и T_{Dt} – это общее время простоя. Задачи распределяются между выбранными, недостаточно загруженными виртуальными машинами в каждом кластере с помощью модели оптимизации. Лучшее решение, найденное с помощью этой модели, должно служить основой для распределения.

Целевая функция Q_{RT} должна определяться исходя из следующего условия:

$$Q_{RT} = \max(T_{vgRT}, W_{RT}, A_{RT}). \quad (4)$$

Система стремится достичь порогового значения производительности, которое определяется максимальным значением среднего времени отклика, пропускной способности и доступности. В этом случае, система хорошо сбалансирована и ни один из показателей производительности не опускается ниже определённого приемлемого уровня.

Распределение рабочей нагрузки между одним или несколькими серверами, сетевыми портами, жёсткими дисками или другими компьютерными ресурсами называется балансировкой нагрузки. Сетевая инфраструктура и мощное (причем дорогостоящее) вычислительное оборудование, используемое в типичных центрах обработки данных, подвержены тем же рискам, что и любое физическое устройство, например сбоям в работе оборудования, перебоям в подаче электроэнергии или работе других сетей, а также нехватке ресурсов в периоды высокого спроса. Балансировка нагрузки в облаке отличается от традиционных теорий проектирования и реализации этой балансировки тем, что для балансировки нагрузки используются общедоступные серверы [2]. Нагрузка может быть перенесена с исходных узлов, на которых выполняется больше работы, на целевые узлы, которые сравнительно менее загружены, для обеспечения сбалансированного распределения задач. Динамическая балансировка нагрузки применяется в режиме реального времени. В зависимости от способа выбора узлов выполнения для достижения динамического распределения нагрузки могут использоваться прямые или итеративные методы.

Предлагаемая работа посвящена балансировке нагрузки между процессорами и виртуальными машинами в облачной среде с помощью гибридных методов машинного обучения с учителем (с целевым атрибутом, то есть с использованием искусственной нейронной сети) и без учителя (без целевого атрибута, то есть с использованием кластеризации методом k-средних) для эффективного расчета нагрузки и кластеризации виртуальных машин. Предлагаемая облачная среда состоит из M процессоров:

$$P = \{P_{M1}, P_{M2}, \dots, P_{Mn}\}.$$

В каждом проекте используется L виртуальных машин V :

$$M = \{V_{M1}, V_{M2}, \dots, V_{Mi}\}.$$

Облачная среда связана с S количеством пользовательских задач, представленных в виде

$$T = \{T_1, T_2, \dots, T_s\}.$$

Чтобы рассчитать нагрузку на каждую виртуальную машину, используется гибридный подход к глубокому обучению, сочетающий свёрточные нейронные сети (СНС) и рекуррентные нейронные сети (РНС) [3].

Свёрточные нейронные сети используются для извлечения объектов из данных датчиков. РНС хорошо подходят для обработки последовательных данных, в частности для сбора временных зависимостей и взаимосвязей между показаниями датчиков с течением времени. Это важно для расчета нагрузки в динамических системах.

Виртуальные машины распределяются по перегруженным и разгруженным кластерам в зависимости от рассчитанной нагрузки.

Для разделения виртуальных машин (ВМ) на перегруженные и разгруженные кластеры в соответствии с рассчитанной нагрузкой используется стратегия на основе пороговых значений нагрузки. Виртуальные машины (ВМ), нагрузка на которые превышает это пороговое значение, считаются перегруженными, а те, нагрузка на которые ниже этого значения, считаются разгруженными. Пороговое значение можно изменить в соответствии с конкретными потребностями и особенностями среды облачных вычислений. Принимая L_i в качестве расчетной нагрузки V_{Mi} , получим выражение для уравнения, которое разделяет виртуальные машины на перегруженные и разгруженные кластеры:

$$V_{MS_перегруженный} = \{i | L_i > L_{порог}\}, \quad V_{MS_недогруженный} = \{i | L_i < L_{порог}\}.$$

Моделирование взаимосвязи между общим агентом и окружающей средой состоит из одного элемента – набора доступных измерений A и функции вознаграждения $S \times A \rightarrow s_k$, как показано на рисунке 1.

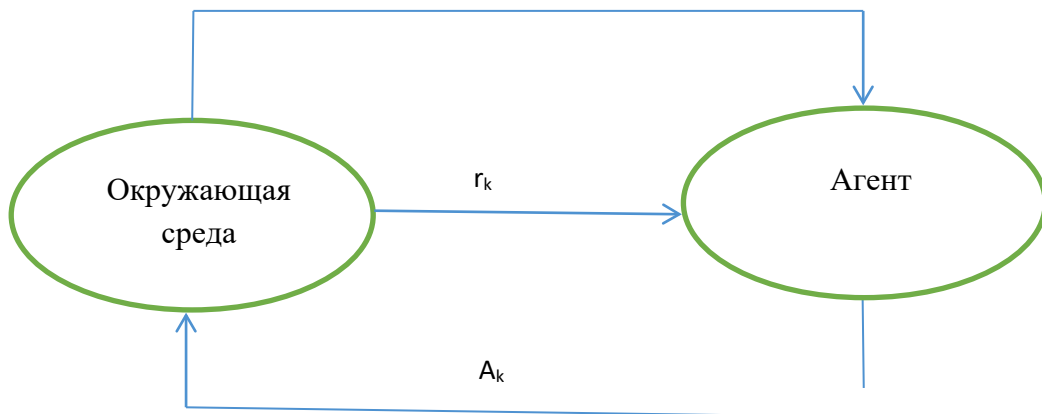


Рис.1. Моделирование взаимосвязи между общим агентом и окружающей средой

Взаимодействие агента с окружающей средой называется непрерывной фазой. На протяжении всего процесса принятия решений

агент основывает свои решения (A_k) о текущем состоянии (S_k) окружающей среды для обновления состояния (S_{k+1}). После принятия решения агент получает информацию о среде для принятия решений в будущем. Далее система должна принять решение и внести необходимые изменения. Агент постоянно пытается максимизировать совокупное вознаграждение, а среда также предоставляет агенту вознаграждение r_k в соответствии с выбором A_k . Для того чтобы агент мог понять, как ему следует вести себя и какие правила соблюдать, должна существовать чёткая система ввода данных. Если система начинает работу в состоянии S_0 и выполняет действие A_0 (и, соответственно, определённую политику), то прогнозируется, что функция ценности $Q(S_i, A_i)$ будет накапливаться (с учётом скидок).

При этом агент пытается максимизировать функцию ценности с помощью Q-обучения.

Функция ставки вознаграждения обозначается как $r(t)$, в то время как ставка дисконтирования представлена в виде β . В Q-обучении используется непрерывная формулировка функции ценности $Q(S, A)$, что указано в уравнении (5):

$$Q(S, A) = E_n \int_{t_0}^{\infty} e^{-\alpha(t_i - t_0)} r(t_i) dt \mid S_0 = S, A_0 = A. \quad (5)$$

Уравнение (5) представляет собой функцию Q-значения в RL, которая оценивает ожидаемое вознаграждение за выполнение действия A в состоянии S . Здесь $Q(S, A)$ вычисляется как ожидаемое значение E_n о совокупном доходе в будущем $r(t_i)$ с коэффициентом дисконтирования $e^{-\alpha(t_i - t_0)}$. Это гарантирует, что более ранние вознаграждения будут иметь больший эффект, чем более поздние, и позволит контролировать влияние будущих вознаграждений [4].

Далее представим правило обновления, вызванное переходом в другое состояние, для времени принятия решений $t+1$, которое определяется следующим образом:

$$eQ^{(k+1)}(S_k, A_k) \leftarrow Q^{(k)}(S_k, A_k) + \left[\frac{1 - e^{-\beta t_k}}{\beta} r(S_k, A_k) + \max e^{-\beta t_k} Q^{(k)}(S_{k+1}, A') Q^{(k)}(S_k, A_k) \right], \quad (6)$$

где $r(S_k, A_k)$ – функция вознаграждения; t_k – время пребывания в состоянии S_k , предшествующем переходу, и $\alpha \leq 1$ – скорость обучения. $Q(S_k, A_k)$ – это оценочное значение решения на каждой итерации t_k .

Уравнение (6) представляет собой правило обновления Q-обучения, которое определяет обновлённую функцию Q-значения в обучении с подкреплением с учётом экспоненциальной корректировки

вознаграждения. Левая часть уравнения $Q^{(k+1)}(S_k, A_k)$ представляет собой обновлённое значение Q для состояния S_k и действия A_k . Правая часть состоит из предыдущего значения Q и коэффициента обучения β , а также коэффициента корректировки вознаграждения, на который влияет фактор экспоненциального затухания. Он также включает в себя функцию максимального значения для возможных действий в следующем состоянии, что обеспечивает оптимальный выбор действия.

Вычитание $Q^{(k)}(S_k, A_k)$ гарантирует, что обновление будет связано с предыдущим значением, поддерживая баланс между новой информацией и уже имеющимися знаниями. Функция полезности оценивается с помощью модифицированного алгоритма Falcon Optimization (MFOA) в ходе автономной процедуры:

$$R = \sum_{t=0}^T \gamma^t r_t. \quad (7)$$

Уравнение (7) представляет собой дисконтированное совокупное вознаграждение в RL. Здесь R – это общее вознаграждение, полученное в результате суммирования непосредственных вознаграждений r_t с течением времени, с учётом коэффициента дисконтирования γ . Коэффициент дисконтирования γ (где $0 \leq \gamma \leq 1$) определяет, в какой степени будущие вознаграждения влияют на общее вознаграждение, при этом меньшие значения отдают приоритет немедленным вознаграждениям.

Орлы обладают уникальным и сложным охотничьим поведением. Они используют как простые, легко распознаваемые приёмы, так и более изощрённые и запутанные тактики, чтобы выследить и поймать добычу. Рассмотрим шаги алгоритма кластеризации на основе метода MFOA.

1. На начальном этапе инициализируются коэффициенты алгоритма оптимизации MFOA и основные ограничения.

2. После этого движение и местоположение орлов изменяются в соответствии с заданными значениями, что выглядит следующим образом:

$$L = \begin{bmatrix} L_{1,1} & \dots & L_{1,V} \\ \dots & \dots & \dots \\ L_{A,1} & \dots & L_{A,D} \end{bmatrix},$$

где L – расположение орла относительно общего числа кандидатов A для каждого измерения V . Скорость генерируется случайным образом в пределах v_{Max} и v_{Min} , которые определяются следующим образом:

$$v_{Max} = 0.1 U_{pli}, \quad v_{Min} = -v_{Max},$$

где U_{pli} – это верхний предел для каждого измерения.

3. Проводится оценка и определение глобальных и индивидуальных оптимальных условий для каждого орла. Получено значение приспособленности, связанное с проблемой, которое формирует вектор. Для определения значения приспособленности на каждой итерации используется целевая функция OF.

$$OF = \begin{bmatrix} of_1 \\ \dots \\ of_n \end{bmatrix}.$$

В этот момент наилучшее индивидуальное местоположение каждого орла обозначается как x_{best} , а наилучшее индивидуальное местоположение фиксируется как g_{best} .

4. На данном этапе оптимальной считается особь I_b , оптимальное положение для каждого орла представим как u_b . Для изучения связи между осознанностью и вероятностью броска создаются два случайных элемента (Q_b, Q_c) с нормальным распределением для каждой атакующей птицы.

5. Оптимальное определение позиции с помощью MFOA с дополненным RL. Гибридный подход сочетает в себе интеллект орла и обучение с подкреплением. Агент RL влияет на принятие решений о вероятности прыжка (Q_r, Q_k), что позволяет более эффективно распределять рабочую нагрузку:

$$L_t = L_{t-1} + V_{t-1} + M_t r (L_{t,r-1} - L_{t-1}) + O_t r (I_{k,t-1} - L_{t-1}). \quad (8)$$

Здесь когнитивный уровень $M_t r$ и социальное взаимодействие $O_t r$ динамически настраиваются с помощью обучения с подкреплением.

6. Обновление местоположения на основе «побега» обеспечивает диверсификацию:

$$L_t = L_{t-1} + \text{rand}(SA_t - R_s L_{t-1}), \quad (9)$$

где SA_t – это безопасная зона, а rand – это контролируемая политикой случайная функция, обеспечивающая исследование.

7. Окончательная оптимизация нагрузки определяется с помощью Falcon на основе RL: на заключительном этапе используется обучение с подкреплением для выбора наиболее эффективной конфигурации планирования задач:

$$L_n = L_t - 1 + V_t - 1 + f_{cr}(L_t - L_{t-1}). \quad (10)$$

Благодаря непрерывному обучению на основе прошлых решений по планированию RL оптимизирует динамическое распределение ресурсов, сокращая количество нарушений SLA и повышая эффективность облачной системы. Благодаря интеграции MFOA с обучением с подкреплением система улучшает распределение задач, оптимизацию ресурсов и динамическую масштабируемость, что приводит к

повышению эффективности работы облака, сокращению количества сбоев и улучшению надежности.

Таблица 1 представляет собой сравнение производительности различных методов при выполнении задач разного объема.

Таблица 1

Сравнение производительности различных методов балансировки нагрузки

Количество задач	RL	LOA	FOA	MFOA
500	84,2	81,5	88,8	89,4
1000	83,9	83,2	88,3	90,6
1500	81,6	82,6	87,7	93,2
2000	81,1	80,4	85,9	94,3
2500	79,7	82,1	85,2	97,6

В таблице 1 представлены четыре методики: RL, LOA, FOA и MFOA. Показатели производительности зафиксированы для пяти различных объемов задач: 500, 1000, 1500, 2000 и 2500 задач. Результаты показывают, что FOA достигает самых высоких показателей производительности при меньшем количестве задач, достигая 88,8 для 500 задач. RL и LOA демонстрируют стабильные тенденции: RL снижается с 84,2 при 500 задачах до 79,7 при 2500 задачах, а LOA колеблется между 83,2 и 80,4. Примечательно, что MFOA демонстрирует наилучшую производительность при выполнении задач любого объема. Эти результаты подтверждают, что разработанный метод обеспечивает наиболее эффективную и масштабируемую производительность среди всех сравниваемых методов, что делает его предпочтительным выбором для обработки больших объемов данных.

Литература

1. Киселев Н.К., Ковтун А.А. Методы обучения нейросетей: метаобучение // Электронный сетевой политематический журнал «Научные труды КубГТУ». 2025. № 5. С. 55-71.
2. Портнов Е.М., Кугоев И.А., Аунг Ч.М. Разработка метода повышения производительности в симметричных многопроцессорных вычислительных системах // Перспективы науки. 2025. № 6 (189). С. 92-98.
3. Димитриченко Д.П. Оптимизация рекуррентной нейронной сети при помощи автоматов с переменной структурой // Программные системы и вычислительные методы. 2023. № 4. С. 30-43.
4. T. Sharma and R.P.S. Bedi, Design and development of pragmatic load balancing algorithm for cloud environment, *Wireless Pers. Commun.*, Vol. 136, No. 1, P. 81–101, May 2024.
5. H. Mikram, S. El Kafhali, and Y. Saadi HEPGA: A new effective hybrid algorithm for scientific workflow scheduling in cloud computing environment, *Simul. Model. Pract. Theory*. Vol. 130, Jan. 2024, Art. No. 102864.

I.A. Kugoev

National Research University of Electronic Technology (MIET)

DEVELOPMENT OF A LOAD BALANCING METHOD IN CLOUD SYSTEMS BASED ON A MODIFIED FALCON OPTIMIZATION ALGORITHM

Keywords: *load balancing, cloud computing, convolutional neural networks, algorithm, reinforcement learning, virtual machines.*

Abstract. *Cloud computing is a scalable and cost-effective platform that provides access to shared computing resources according to the user's request. Effective load balancing is necessary to achieve optimal performance and maximize resource utilization, ensuring an even distribution of network traffic between servers, preventing congestion, increasing response time, and increasing system reliability. This article proposes a method for load balancing in cloud systems based on a modified Falcon Optimization algorithm and reinforcement learning. The proposed model uses convolutional and recurrent neural networks to predict the load on virtual machines and dynamically classify them into categories of congestion and underloading. The proposed approach significantly improves task planning and load balancing compared to existing methods such as the load optimization algorithm (LOA), reinforcement learning (RL) and the Falcon optimization algorithm (FOA).*

В.А. Лопырёв, Л.Г. Гагарина

Национальный исследовательский университет «МИЭТ»

УДК 004.657

РАЗРАБОТКА АЛГОРИТМА ПАРАЛЛЕЛЬНОГО СКАНИРОВАНИЯ СЖАТЫХ КОЛОНОЧНЫХ ДАННЫХ С ДЕКОМПРЕССИЕЙ И ДЕКОДИРОВАНИЕМ

Ключевые слова: *колоночные данные, аналитические СУБД, OLAP, распределенные системы, сжатие данных, декодирование, параллельное выполнение, многопоточность, кеш, векторизация.*

В статье рассматривается проблема производительности операции сканирования сжатых и закодированных колоночных данных. Основное внимание уделено повышению производительности чтения данных за счет эффективного использования многопоточности, векторизации и кеш-ориентированности. Предложен алгоритм, объединяющий декомпрессию страниц памяти, декодирование

колоночных представлений, векторизованную обработку предикатов и планировщик для равномерного распределения нагрузки на ядра центрального процессора. Эффективность представленного решения подтверждается экспериментально.

Колоночные форматы хранения данных являются основой аналитических СУБД благодаря высокой эффективности чтения по столбцам и возможности применять методы сжатия и кодирования, учитывая особенности типов данных. Основное преимущество колоночного подхода – уменьшение объема читаемых данных и рост эффективности кешей и предвыборки при выполнении аналитических запросов [1].

Однако практическая производительность сканирования колоночных данных часто ограничивается не только пропускной способностью жесткого диска, но и стоимостью «распаковки» данных на CPU. В современных форматах (Parquet, ORC) распространён двухуровневый подход:

- 1) декомпрессия (zstd, snappy, gzip) – восстановление байтовых блоков и страниц;
- 2) декодирование (dictionary, RLE, bit-packing, delta) – восстановление логического представления столбца.

Задача параллельного сканирования осложняется тем, что декомпрессия и декодирование:

- создают нерегулярные ветвления;
- активно потребляют память;
- конкурируют за кеш и пропускную способность памяти;
- плохо масштабируются без продуманной стратегии распределения работы по потокам и NUMA-локальности.

Цель настоящей статьи – разработать алгоритм параллельного сканирования сжатых колоночных данных, включающий декомпрессию и декодирование, векторизацию и многопоточную модель выполнения.

Анализ существующих решений. Классические модели планирования и выполнения запросов, обрабатывающие по строке за раз, страдают от высокой стоимости вызовов операторов и слабого использования возможностей современных CPU. Векторизованный подход позволяет обрабатывать блоки значений фиксированного размера, что улучшает локальность и снижает накладные расходы интерпретации. В MonetDB запросы выполняются в формате «вектор за вектором», где каждый вектор содержит порядка тысячи значений, что демонстрирует преимущества такого конвейера. DuckDB также использует векторизацию и фиксированный размер вектора по умолчанию (порядка тысячи значений), что снижает накладные расходы вычислений и упрощает буферизацию.

Однако векторизация сама по себе не гарантирует эффективности декомпрессии и декодирования, если они выполняются как отдельные, недостаточно распараллеленные стадии или приводят к избыточной материализации.

Для многоядерных систем важны как многопоточность, так и управление локальностью памяти. Morsel-driven framework предлагает дробить входные данные на небольшие кусочки (morsels) и планировать их исполнение как небольшие задачи, учитывая локальность расположения данных в памяти, процессорный кеш и динамическую балансировку нагрузки [2].

В то же время типовые описания morsel подхода фокусируются на реляционных операторах, но не детализируют оптимальное слияние декомпрессии, декодирования и фильтрации для современных колоночных форматов с разными схемами кодирования.

Колоночный формат Parquet широко использует словарное кодирование и хранит индексы словаря как целые числа, закодированные RLE или bit-packing. При росте словаря предусмотрено отключение кодирования из-за падения его эффективности [3]. ORC специфицирует набор алгоритмов кодирования: dictionary, bit-packing, delta, RLE и допускает дополнительную общую компрессию поверх них [4].

Таким образом, алгоритм сканирования должен уметь быстро переключаться между схемами декодирования на уровне страниц памяти и максимально использовать векторизацию на целочисленных массивах данных.

Декодирование упакованных целых чисел занимает заметную часть CPU-времени. Векторизация позволяет экономить процессорное время, а схемы семейства SIMD-BP128 и SIMD-FastPFOR дают кратный прирост скорости декодирования относительно скалярных подходов [5].

Однако даже очень быстрый SIMD-декодер может упереться в пропускную способность памяти или в стоимость последующей фильтрации и материализации, если конвейер, выполняющий запросы, не оптимизирован целиком.

Алгоритм параллельного сканирования сжатых колоночных данных. Рассматривается колоночный формат хранения данных, разбитых на группы строк, а внутри – на колоночные сегменты, состоящие из страниц. Каждая страница содержит:

- метаданные (тип сжатия, кодирования, статистики / минимумы-максимумы, nullable-маска);
- байтовый блок, сжатый общим алгоритмом сжатия;
- внутри байтового блока – массив значений, закодированный подходящей схемой (dictionary, RLE, bit-packing, delta).

Необходимо реализовать операцию сканирования для запросов с выбором подмножества столбцов (проекция) и предикатами (фильтрами) по части столбцов.

Пусть C – множество столбцов данных, d – количество столбцов:

$$C = \{c_1, \dots, c_d\}.$$

Пусть S – множество столбцов проекции:

$$S \subseteq C.$$

Пусть F – предикат, D_c – домен значений столбца c (включая $NULL$):

$$F: \prod_{c \in C} D_c \rightarrow \{0, 1\}$$

Пусть $C_S(F)$ – столбцы, необходимые для выполнения фильтрации данных:

$$C_S(F) \subseteq C.$$

Разделение строк столбца на блоки имеет вид:

$$[0, N) = \bigcup_{j=0}^{m-1} R_j, \quad R_j = [k_j, u_j), \quad n_j = u_j - k_j,$$

где N – число строк, R_j – диапазон строк j -го блока, k_j и u_j – начало и конец диапазона, n_j – число строк в блоке, m – число блоков.

Для каждого столбца c в блоке j задан физический фрагмент:

$$P_{cj} = \langle B_{cj}^{cmp}, cdc_{cj}, enc_{cj}, M_{cj} \rangle,$$

где B_{cj}^{cmp} – сжатые байты фрагмента, cdc_{cj} – алгоритм общей компрессии, enc_{cj} – алгоритм логического кодирования значений, M_{cj} – метаданные.

Функции декомпрессии и декодирования имеют вид:

$$B_{cj} = decompress(B_{cj}^{cmp}, cdc_{cj}),$$

где B_{cj} – распакованный байтовый блок;

$$v_{cj} = decompress(B_{cj}, enc_{cj}) \in D_c^{n_j},$$

где v_{cj} – массив значений столбца c для блока j , $D_c^{n_j}$ – последовательность длины n_j из домена D_c .

Пусть W – размер пакета для векторизации, q_j – число пакетов в блоке j , тогда:

$$q_j = \left\lceil \frac{n_j}{W} \right\rceil.$$

Тогда диапазон индексов пакета:

$$I_{jk} = [kW, \min((k+1)W, n_j)],$$

где $k \in \{0, \dots, q_j - 1\}$, I_{jk} – диапазон локальных индексов внутри блока j .

Векторное декодирование этого диапазона:

$$v_{cjk} = \text{decode_vec}(B_{cj}, \text{enc}_{cj}, I_{jk}) \in D_c^{|I_{jk}|},$$

где v_{cjk} – значения столбца c в пакете (j, k) , $|I_{jk}|$ – длина пакета.

Фильтрация значений по заданным условиям имеет вид:

$$s_{jk}[p] = F((v_{cjk}[p])_{c \in C_S(F)}) \in \{0, 1\},$$

где $p \in [0, |I_{jk}|]$ – позиция внутри пакета, $c \in C_S(F)$ означает набор значений столбцов, необходимых предикату.

Результат обработки пакета значений после проекции и фильтрации:

$$O_{jk} = ((v_{cjk}[p])_{c \in S} : s_{jk}[p] = 1),$$

где $(v_{cjk}[p])_{c \in S}$ – кортеж из столбцов проекции S .

Результат сканирования:

$$\text{Scan}(S, F) = \bigoplus_{j=0}^{m-1} \bigoplus_{k=0}^{q_j-1} O_{jk},$$

где $\bigoplus$ – конкатенация пакетов.

Для ускорения выполнения операции сканирования применяется поздняя материализация:

$$C_n = S \cup C_S(F),$$

где C_n – минимально необходимое множество столбцов, $C_S(F)$ декодируется для построения s_{jk} , $S \setminus C_S(F)$ декодируется после построения s_{jk} .

Для параллельного сканирования определим множество задач по принципу «одна задача – один пакет одного блока»:

$$T = \{t_{jk} \mid j \in [0, m), k \in [0, q_j)\}$$

Назначим задачи потокам:

$$\sigma : T \rightarrow \{0, \dots, N_T - 1\},$$

где N_T – число потоков, $\sigma(t_{jk})$ – номер потока, который обрабатывает пакет (j, k) .

Время выполнения:

$$T(\sigma) = \max_{i \in [0, N_T - 1]} \sum_{\tau \in T: \sigma(\tau) = i} C(\tau),$$

где $C(\tau)$ – стоимость задачи, измеряемая временем и равная сумме времени, потраченного на декомпрессию, декодирование, фильтрацию и проекцию, функция $\max$ выбирает самый загруженный поток.

Таким образом, цель планирования при распараллеливании работы: $\min_{\sigma} T(\sigma)$.

Экспериментальная оценка эффективности предложенного решения. Экспериментальная оценка эффективности проводилась по двум критериям: корректности и скорости выполнения запроса.

Для подтверждения корректности необходимо доказать, что

$$O_{prop}(S, F) = O_{ref}(S, F),$$

где $O_{prop}(S, F)$ – результат выполнения запроса, используя предложенный алгоритм, $O_{ref}(S, F)$ – эталонный результат.

Для многопоточного режима ($N_T > 1$) корректность определяется по следующей формуле:

$$O_{N_T=1}^{prop} = O_{N_T=k}^{prop},$$

где $O_{N_T=1}^{prop}$ – результат выполнения запроса в однопоточном режиме, $O_{N_T=k}^{prop}$ – результат выполнения запроса в многопоточном режиме, используя k потоков.

Для определения эффективности алгоритма используются следующие критерии:

- время T и пропускная способность Q :

$$Q_r = \frac{N}{T} \left(\frac{\text{строк}}{c} \right) \quad Q_v = \frac{N * c}{T} \left(\frac{\text{значений}}{c} \right),$$

где N – количество строк в колонке, c – количество декодированных и материализованных колонок;

- масштабирование:

$$Sp(N_T) = \frac{T(1)}{T(N_T)} \quad E(N_T) = \frac{Sp(N_T)}{N_T} = \frac{T(1)}{N_T * T(N_T)},$$

где $Sp(N_T)$ – ускорение при использовании N_T потоков, $E(N_T)$ – эффективность.

Определив критерии, выполним сравнение следующих алгоритмов сканирования:

- 1) однопоточное выполнение, включающее последовательные отдельные этапы декодирования, полной материализации данных и фильтрации;
- 2) двухпоточное выполнение, включающее последовательные отдельные этапы декодирования, полной материализации данных и фильтрации;
- 3) четырехпоточное выполнение, включающее последовательные отдельные этапы декодирования, полной материализации данных и фильтрации;
- 4) четырехпоточное выполнение, включающее этапы morsk-планирования и декодирования, объединенного с фильтрацией, а также с применением поздней материализации данных.

Все перечисленные алгоритмы сканирования используют векторизацию для декодирования данных.

Эксперимент проводился на тестовом наборе данных, состоящем из 5 колонок, каждая из которых содержит 10 млн элементов типа INT8 или int64 – целочисленного типа, занимающего 8 байт. Предикат для фильтра задан по первой колонке на 50% значений.

Результаты оценки эффективности алгоритмов сканирования представлены на рисунке 1. В представленных временных показателях не учитывается время, затрачиваемое на операции чтения данных с жесткого диска, так как они вносят дополнительную погрешность в измерения.

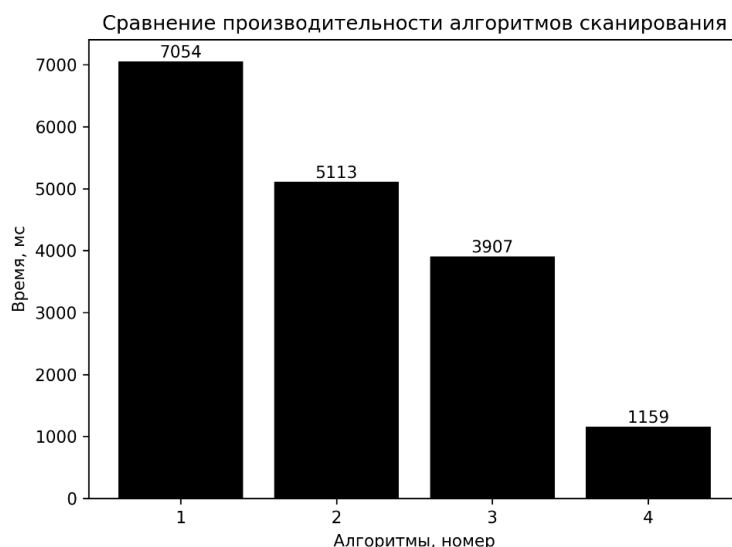


Рис. 1. Сравнение производительности алгоритмов сканирования

Критерий корректности для одного и нескольких потоков выполняется для всех протестированных алгоритмов.

Для первого алгоритма получили:

$$Q_{r1} = \frac{10000000}{7,054} \cong 1\,417\,635 \left(\frac{\text{строк}}{\text{с}} \right)$$

$$Q_{v1} = \frac{10000000 * 5}{7,054} \cong 7\,088\,177 \left(\frac{\text{значений}}{\text{с}} \right)$$

$$Sp_1(1) = \frac{7,054}{7,054} = 1$$

$$E_1(1) = \frac{1}{1} = 1$$

Для второго алгоритма получили:

$$Q_{r2} = \frac{10000000}{5,113} \cong 1\,955\,799 \left(\frac{\text{строк}}{\text{с}} \right)$$

$$Q_{v2} = \frac{10000000 * 5}{5,113} \cong 9\,778\,995 \left(\frac{\text{значений}}{\text{с}} \right),$$

$$Sp_2(2) = \frac{7,054}{5,113} \cong 1,38$$

$$E_2(2) = \frac{1,38}{2} = 0,69$$

Для третьего алгоритма получили:

$$Q_{r3} = \frac{10000000}{3,907} \cong 2\,559\,231 \left(\frac{\text{строк}}{\text{с}} \right)$$

$$Q_{v3} = \frac{10000000 * 5}{3,907} \cong 12\,797\,543 \left(\frac{\text{значений}}{\text{с}} \right),$$

$$Sp_3(4) = \frac{7,054}{3,907} \cong 1,81$$

$$E_3(4) = \frac{1,81}{4} \cong 0,45$$

Для предложенного алгоритма получили:

$$Q_{r4} = \frac{10000000}{1,159} \cong 8\,628\,128 \left(\frac{\text{строк}}{\text{с}} \right)$$

$$Q_{v4} = \frac{10000000 * 1}{1,159} \cong 8\,628\,128 \left(\frac{\text{значений}}{\text{с}} \right),$$

$$Sp_4(4) = \frac{7,054}{1,159} \cong 6,09 \quad E_4(4) = \frac{6,09}{4} \cong 1,52$$

Самым медленным является первый алгоритм с одним потоком выполнения, отдельными операциями декодирования и фильтрации, а также полной материализацией данных. Использование двух и четырех потоков позволило сократить время сканирования на 28% и 45% соответственно. Наилучший результат показал предложенный алгоритм, ускорив выполнение операции в 6 раз относительно первого алгоритма.

Выводы. В статье рассмотрена проблема параллельного сканирования сжатых колоночных данных, заключающаяся в дорогих операциях декомпрессии и декодирования, оказывающих повышенное давление на память и кеш. На основе анализа векторизованных движков, morsel-driven планирования и особенностей колоночных форматов Parquet и ORC предложен алгоритм, объединяющий операции декодирования и фильтрации, векторизующий обработку целочисленных типов данных, поддерживающий отложенную материализацию с учетом маски выбора и планировщик, обеспечивающий параллельную обработку данных. Представленный подход направлен на снижение объемов данных, которые центральный

процессор перемещает через иерархию памяти во время сканирования, а также устранение лишней материализации, что особенно важно при высоких степенях параллелизма.

Литература

1. Columnar databases explained. // clickhouse: [сайт]. URL: <https://clickhouse.com/resources/engineering/what-is-columnar-database> (дата обращения: 27.02.2026).
2. Viktor Leis, Peter Boncz, Alfons Kemper, Thomas Neumann. Morsel-driven parallelism: a NUMA-aware query evaluation framework for the many-core age // In Proceedings of the 2014 International Conference on Management of Data (SIGMOD '14). 2014. P. 743-754.
3. File Format // parquet.apache: [сайт]. URL: <https://parquet.apache.org/docs/file-format/> (дата обращения: 27.02.2026).
4. ORC Specification v1. // orc.apache. URL: <https://orc.apache.org/specification/ORCv1/> (дата обращения: 27.02.2026).
5. Lemire D, Boytsov L. Decoding billions of integers per second through vectorization // Software: Practice & Experience. 2015. Vol. 45, №1. P. 1-29.

V.A. Lopyryov, L.G. Gagarina

National Research University of Electronic Technology (MIET)

Keywords: *columnar data, analytical DBMS, OLAP, distributed systems, data extension, decoding, parallel execution, multithreading, cache, vectorization.*

Abstract. *This paper examines the performance of scanning compressed and encoded columnar data. The focus is on improving data read performance through the efficient use of multithreading, vectorization, and cache-awareness. An algorithm is proposed that combines memory page decompression, columnar representation decoding, vectorized predicate processing, and a scheduler for uniform load distribution across CPU cores. The effectiveness of the proposed solution is confirmed experimentally.*

М.В. Масленников

*Московский государственный университет
им. Ломоносова*

УДК 004.032.26

СКВОЗНОЙ (λ, μ) -АНАЛИЗ И СТРУКТУРНОЕ РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ В АРХИТЕКТУРЕ МАМБА

Ключевые слова: (λ, μ) -формализм, многомерные матрицы, Мамба, SSM, параллельные вычисления, тензорный параллелизм.

В статье предлагается новый теоретический подход к распараллеливанию вычислений в архитектуре State Space Models (SSM), в частности Mamba, основанный на (λ, μ) -формализме многомерных матриц Соколова. В отличие от существующих инженерных решений, оптимизирующих отдельные матричные операции, мы вводим концепцию сквозного анализа потока данных через всю сеть на основе классификации индексов на скоттовы, кэлиевы и свободные. Мы показываем, что такое представление позволяет математически обосновать минимальный объём необходимой синхронизации градиентов и предложить метод оптимального распределения вычислений по множеству GPU. Работа носит теоретический характер и закладывает фундамент для создания автоматического планировщика параллельных вычислений нового поколения.

1. Введение. Современные архитектуры глубоких нейросетей, такие как Mamba [1], демонстрируют выдающиеся результаты в обработке последовательностей, однако их обучение на многопроцессорных кластерах остаётся сложной инженерной задачей. Ключевое узкое место – коммуникация при синхронизации градиентов, объём которой в существующих подходах пропорционален размерности модели и достигает миллионов чисел.

Существующие фреймворки распределённого обучения (TensorFlow [3], NeMo [2]) используют эвристические стратегии, не учитывающие внутреннюю алгебраическую структуру тензоров. Это приводит к избыточной синхронизации и требует ручной настройки.

В данной работе мы предлагаем принципиально иной подход, основанный на (λ, μ) -формализме многомерных матриц, развитом в работах Соколова [8] и Мунермана [4]. Наша гипотеза заключается в том, что анализ типов индексов в тензорах, проходящих через всю сеть, позволяет:

- 1) математически доказать минимально возможный объём синхронизации;
- 2) структурно распараллелить вычисления, разрезая тензоры по естественным линиям раздела;
- 3) заложить основу для автоматического планировщика параллельных вычислений.

Цель данной статьи – изложить концепцию сквозного (λ, μ) -анализа на примере архитектуры Mamba и сформулировать постановку задачи для последующих исследований и разработок.

2. (λ, μ) -формализм: типология индексов. В основе нашего подхода лежит представление тензоров как многомерных структур, где каждый индекс имеет не только координатное, но и алгебраическое

значение. При умножении двух тензоров индексы проявляют себя тремя различными способами.

Определение 1 (типы индексов). Рассмотрим умножение двух тензоров A и B , в результате которого получается тензор C . Можно различить следующие типы индексов:

1. **Скоттовы индексы (S).** Присутствуют в обоих тензорах A , B и остаются в тензоре-результате C . Они являются "мостом", связывающим структуры, но не участвуют во внутренней свёртке.

2. **Кэлиевы индексы (K).** Присутствуют в обоих тензорах A , B , но исчезают (суммируются) в процессе умножения, не переходя в результат C . Они являются "клеем", по которому происходит свёртка.

3. **Свободные индексы (F).** Присутствуют только в одном из тензоров A , B и переходят в результат C . Они являются «хвостами», которые просто переносятся через операцию.

Эта классификация является фундаментальной и зависит не от конкретных числовых значений тензоров, а только от формы тензоров и выполняемой операции.

Пример 1. Умножение матрицы на вектор. Для операции $y[i] = \sum_j W[i, j]x[j]$: индекс i – свободный (присутствует в W и в результате y), индекс j – кэлиев (присутствует в W и x , суммируется).

Пример 2. Умножение двух матриц. Для операции $C[i, k] = \sum_j A[i, j, k] \cdot B[j, k]$: индексы i и k – свободные, индекс j – кэлиев, индекс k – скоттов.

3. Архитектура Mamba в терминах типов индексов. Применим данный формализм к ключевым блокам архитектуры Mamba [1; 5]. Рассмотрим стандартный блок Mamba с параметрами: $batch$, seq , d_{model} , $expand$ (обычно 2), d_{state} (обычно 16). В архитектуре Mamba после проекции вверх размерность увеличивается: $d_{inner} = expand \cdot d_{model}$, где $expand$ обычно равен 2. Далее все внутренние операции SSM и свёртки работают в пространстве размерности d_{inner} , после чего проекция вниз возвращает исходную размерность d_{model} .

3.1. Линейная проекция вверх. Операция умножения $X \cdot W_{up}$, где $X: [batch, seq, d_{model}]$, $W_{up}: [d_{model}, 2 \cdot d_{inner}]$. Свободными индексами здесь являются $batch$, seq , а также новая размерность $expand \cdot d_{model}$. Кэлиев индекс — размерность d_{model} , по которой идёт суммирование.

3.2. Селективное обновление состояния. В этой секции приводится полная система уравнений блока Mamba. Сначала с помощью проекций в блоке Mamba вычисляются параметры B_t , C_t и шаг дискретизации Δ_t , зависящие от входа x_t :

Шаг 1. Вычисление параметров, зависящих от входа в SSM

№	Операция	Размерности op_1, op_2	F	S	K	Результат
1	$d_t = W_1 \cdot x_t$	$[dt_{rank}, d_{inn}] \times [B, S, d_{inn}]$	B, S, dt_{rank}		d_{inn}	$[B, S, dt_{rank}]$
2	$\Delta_{raw} = W_2 \cdot d_t + b_\Delta$	$[d_{inn}, dt_{rank}] \times [B, S, dt_{rank}]$	B, S, d_{inn}		dt_{rank}	$[B, S, d_{inn}]$
3	$\Delta_t = \text{softplus}(\Delta_{raw})$	$[B, S, d_{inn}]$	B, S, d_{inn}			$[B, S, d_{inn}]$
4	$B_t = W_B \cdot x_t + b_B$	$[d_{inn}, d_{st}] \times [B, S, d_{inn}]$	B, S, d_{st}		d_{inn}	$[B, S, d_{st}]$
5	$C_t = W_C \cdot x_t + b_C$	$[d_{inn}, d_{st}] \times [B, S, d_{inn}]$	B, S, d_{st}		d_{inn}	$[B, S, d_{st}]$

В таблицах для этого и последующих шагов для краткости используется обозначение размера батча $B = batch$, длины входной последовательности $S = seq$, количества каналов $d_{inn} = d_{inner}$ и размерности внутреннего состояния $d_{st} = d_{state}$. В колонках показаны свободные F , скоттовы S и кэлиевы K индексы, а также размерности результата в левой части.

Шаг 2. Дискретизация

№	Операция	Размерности op_1, op_2	F	S	K	Результат
1	$A \triangleq \exp(\Delta \triangle A)$	$[B, S, d_{inn}] \times [d_{inn}, d_{st}, d_{st}]$	B, S, d_{st}		d_{inn}	$[B, S, d_{inn}, d_{st}, d_{st}]$
2	$B \triangleq \Delta \triangle B$	$[B, S, d_{inn}] \times [d_{inn}, d_{st}]$	B, S, d_{st}		d_{inn}	

На этом и последующих шагах в каждом ядре SSM блока Mamba реализуется рекуррентное обновление скрытого состояния, всего таких каналов d_{inner} . Параметр Δ_t в формулах дискретизации зависит от входа x_t , моделируя важность информации от x_t . Матрица A не зависит от входа и инициализируется отдельно.

Шаг 3. Рекуррентное обновление состояния

№	Операция	Размерности op_1, op_2	F	S	K	Результат
1	$h_t = A \triangle h_{t-1} + B_t$	$[d_{inn}, d_{st}, d_{st}] \times [B, S, d_{inn}, d_{st}]$	B, S, d_{st}	d_{inn}	d_{st}	$[B, S, d_{inn}, d_{st}]$

На этом шаге обновляется состояние в зависимости от предыдущего состояния и текущего входа x_t .

Шаг 4. Чтение состояния в выход

№	Операция	Размерности op_1, op_2	F	S	K	Результат
1	$y_t = C_t \cdot h_t$	$[d_{inn}, d_{st}] \times [B, S, d_{inn}, d_{st}]$	B, S, d_{st}	d_{inn}	d_{st}	$[B, S, d_{inn}]$

Из вышеуказанных формул видно, что после этапа проекций, в котором d_{inner} выступает как кэлиев индекс и требует синхронизации, вычисления происходят независимо по каналам. На этапах дискретизации, рекуррентного обновления и чтения состояния d_{inner}

выступает как свободный или скоттов индекс, что позволяет полностью распараллелить эти вычисления по каналам.

3.3. Поэлементное умножение ветвей. После разделения входного тензора одна ветвь проходит через SSM-ядро, другая – через свертку и активацию SiLU. Затем выполняется поэлементное умножение результатов: $result = y_t * Conv_{out}$, где оба операнда имеют размерность $[batch, seq, d_{inner}]$. В этой операции все индексы ($batch$, seq , d_{inner}) присутствуют в обоих операндах и переходят в результат, то есть являются скоттовыми индексами. Это принципиально важный момент: в поэлементных операциях синхронизация не требуется в принципе.

3.4. Линейная проекция вниз. Операция $result \cdot W_{down}$, где $result : [batch, seq, d_{inner}]$, $W_{down} : [d_{inner}, d_{model}]$. В ней участвуют свободные индексы $batch$, seq и d_{model} как новая размерность, а также кэлиев индекс d_{model} из $result$, по которому идёт суммирование.

Наблюдение 1. Индексы $batch$ и seq являются свободными (а в поэлементном умножении – скоттовыми) на протяжении всей сети. Они никогда не становятся кэлиевыми. Свободные и кэлиевы индексы меняются местами на границах блоков. Эти переходы создают естественные точки синхронизации – места, где тип индекса меняется, и коммуникация между устройствами становится неизбежной. Это наблюдение является ключом к сквозной оптимизации.

4. Следствия для распараллеливания. Проведённый анализ позволяет сформулировать принципы оптимального распараллеливания вычислений в Mamba. Рассмотрим их.

Принцип распределения. Наличие свободных индексов $batch$ и seq во всех операциях позволяет с самого начала распределить вычисления по разным GPU, разрезая тензоры именно по этим размерностям. Такое разбиение сохраняется на протяжении всего прямого и обратного прохода, поскольку данные индексы никогда не становятся кэлиевыми.

Принцип синхронизации. Синхронизация (операция AllReduce) требуется только по кэлиевым индексам. В проекциях это размерность d_{model} , в SSM-ядре – размерность d_{state} . Для типичных параметров Mamba-1.4B ($d_{model} = 2048$, $d_{state} = 16$) объём синхронизируемых данных составляет 2048 чисел в проекциях и 256 чисел в SSM-ядре. Это на несколько порядков меньше объёма данных, которые пришлось бы синхронизировать при наивном подходе (миллионы чисел).

Принцип локализации. Поэлементное умножение ветвей, где все индексы являются скоттовыми, вообще не требует синхронизации. Результаты, полученные на разных GPU, могут быть склеены без какой-либо коммуникации. Это означает, что данный этап вычислений идеально масштабируется на произвольное число устройств.

Итоговая оценка. При предложенном (λ, μ) -подходе объём синхронизации составляет $O(d_{model} + d_{state}^2)$ и не зависит от $batch$ и seq . Для Mamba-1.4B с $batch = 32$ и $seq = 2048$ это даёт сокращение коммуникации в десятки тысяч раз¹ по сравнению с наивным подходом, синхронизирующим все градиенты.

5. Теорема о минимальной синхронизации. Классификация индексов позволяет перейти к количественной оценке минимально необходимого объёма коммуникации. Для этого введём понятие сечения тензора.

Определение 2. Сечением тензора называется подтензор, полученный фиксацией значений всех индексов из заданного множества.

Ключевая идея распараллеливания:

1. По скоттовым и свободным индексам тензор можно разрезать на независимые сечения. Вычисления над каждым таким сечением могут быть выполнены на отдельном вычислительном устройстве (GPU) без какой-либо коммуникации.

2. Синхронизация требуется только по кэлиевым индексам, так как именно там происходит сложение результатов из разных сечений (операция AllReduce).

Теорема 1 (О минимальной синхронизации). Пусть тензор T размера $D_1 \times \dots \times D_n$ участвует в операции свёртки, где множество индексов I разбито на свободные F , скоттовы S и кэлиевы K . При распределении вычислений по P устройствам путём разрезания тензора по индексам из $F \cup S$ объём данных V , необходимых для синхронизации, удовлетворяет $V = \prod_{K \times K} D_i$ на каждое устройство.

Доказательство. В прямой свёртке каждый элемент результата получается суммированием по кэлиевым индексам произведений соответствующих элементов операндов. При обратном распространении градиенты по операндам также требуют суммирования по кэлиевым индексам вкладов от всех свободных и скоттовых сечений. Поскольку свободные и скоттовы индексы распределены по устройствам, для получения глобальных градиентов необходимо сложить частичные суммы, каждая из которых имеет размер, равный произведению размерностей кэлиевых индексов. Никакие другие данные не требуют синхронизации, так как вычисления по свободным и скоттовым индексам независимы.

Следствие 1. Для линейной проекции в Mamba ($d_{model} = 2^{11}$) объём синхронизируемых данных составляет $2^{11} = 2048$ чисел.

При наивной полной синхронизации градиентов, не учитывающей структуру индексов, объём синхронизации пропорционален общему

¹ $batch * seq * d_{model} = 32 \cdot 2048 \cdot 2048 \approx 134$ млн, $2048 + 256 \approx 2300$. 134 млн / $2300 \approx 58\,000$ раз

числу градиентов (~ 4 миллионов). Выигрыш составляет 2000 раз и является следствием фундаментального алгебраического свойства операции, а не инженерной оптимизации.

6. Заключение и дальнейшие исследования. В данной работе представлена концепция применения (λ, μ) -формализма для структурного распараллеливания нейросетей на примере архитектуры Mamba. Мы показали, что классификация индексов на скоттовы, кэлиевы и свободные позволяет математически обосновать минимальный объём необходимой синхронизации и закладывает теоретический фундамент для создания автоматического планировщика параллельных вычислений.

Полученные результаты открывают несколько направлений для дальнейших исследований:

1. Автоматизация анализа. Требуется разработка алгоритмов, способных автоматически определять типы индексов в произвольных вычислительных графах и строить оптимальные планы синхронизации.

2. Реализация и валидация. Необходима программная реализация ключевых операций с учётом предложенной классификации индексов и экспериментальная проверка полученных теоретических оценок.

3. Обобщение на другие архитектуры. Представляет интерес применение (λ, μ) -анализа к трансформерам, где операция внимания создаёт более сложные взаимодействия индексов, и к графовым нейронным сетям, в которых индексы соответствуют вершинам и рёбрам графа.

Благодарности. Автор выражает глубокую благодарность профессору Виктору Иосифовичу Мунерману за плодотворные обсуждения и предоставленные материалы по (λ, μ) -формализму.

Литература

1. Gu A., Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. 2024 // First conference on language modeling.
2. NVIDIA. Mamba2 and Hybrid Models.. 2024. NVIDIA Developer Documentation. URL: <https://docs.nvidia.com/nemo-framework/user-guide/24.07/llms/mamba/index.html>.
3. TensorFlow. Distributed Training with TensorFlow. 2024. TensorFlow Core Documentation. URL: https://www.tensorflow.org/guide/distributed_training.
4. Мунерман В.И. Массовая обработка данных. Алгебраические модели и методы: монография. М.: ИНФРА-М, 2023. 229 с.
5. Dao T., Gu A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. 2024 // arXiv preprint arXiv:2405.21060.
6. Oseledets I.V. Tensor-train decomposition. 2011 // SIAM Journal on Scientific Computing.
7. GitHub Issue #10966. Converting Mamba to tp4: RuntimeError. NVIDIA/NeMo repository.

8. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.

M.V. Maslennikov

Lomonosov Moscow State University

END-TO-END (λ, μ) -ANALYSIS AND STRUCTURAL PARALLELIZATION OF COMPUTATIONS IN THE MAMBA ARCHITECTURE

Keywords: (λ, μ) -formalism, multidimensional matrices, Mamba, SSM, parallel computing, tensor parallelism.

Abstract. *This paper presents a new theoretical approach to parallelizing computations in State Space Models (SSM), in particular Mamba, based on the (λ, μ) -formalism of multidimensional matrices developed by Sokolov and Munerman. Unlike existing engineering solutions that optimize individual matrix operations, we introduce the concept of end-to-end analysis of data flow through the entire network based on the classification of indices into Scott, Cayley, and free types. We show that such representation allows us to mathematically justify the minimum required volume of synchronized data (proportional to the number of Cayley indices). The work is theoretical in nature and lays the foundation for creating a new generation of automatic parallel computing schedulers.*

А.С. Машенков, В.И. Мунерман

Смоленский государственный университет

УДК 004.421.2

ОПТИМИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ПРОИЗВЕДЕНИЯ МНОГОМЕРНЫХ МАТРИЦ

Ключевые слова: многомерные данные, тензоры, многомерные матрицы, свернутое произведение, оптимизация вычислений, последовательность умножения матриц, параллельные вычисления, многомерный анализ данных, обработка данных.

В статье рассматриваются методы представления и обработки многомерных данных с использованием тензоров и многомерных матриц. Показана роль многомерных структур данных в задачах анализа и хранения информации, а также их применение в системах аналитики и поддержки принятия решений. Описаны свойства многомерных матриц и операции над ними, в том числе свернутое произведение. Рассмотрена задача оптимизации последовательности

умножения многомерных матриц, сложность которой существенно зависит от порядка выполнения операций. Предложен подход к сокращению числа допустимых последовательностей на основе учета общих индексов и предотвращения разрыва вычислительной последовательности. Также рассматривается возможность использования полученных результатов для повышения эффективности алгоритмов обработки многомерных данных и организации параллельных вычислений.

Введение. В связи с увеличением объема информации, важно эффективно хранить и обрабатывать данные. Многомерное представление и обработка данных – одна из актуальных тем, которая является главным фактором развития различных областей науки и техники. Такие представления учитывают сложные взаимосвязи между данными, позволяя моделировать различного рода процессы, увеличивая точность их результатов.

Для этого данные организуют в виде многомерных структур таких как OLAP-кубы. Это решение широко используется в системах аналитики, поддержки принятия решений. Такой подход рассматривается в работе [1], где описывается создание многомерной базы данных из различных источников и подготовка отчетов с помощью OLAP-анализа, формируя базы знаний для принятия решений. Также в работе [2] предложен метод многомерного анализа данных с использованием не доопределенных вычислений, позволяющий учитывать неточность исходных данных. Это направление активно развивается, предлагая такие новые подходы хранения, как OLAP и MOLAP, рассмотренные в работе [3].

Еще одним представлением многомерных данных являются тензоры, которые широко используются в различных областях. В работе А. А. Банина [4] показано, как тензоры применяются для анализа многомерных данных с категоризованными признаками. Анализ тензорных сигналов позволяет решать задачи пространственного позиционирования [5]. Также тензоры применяются в анализе сетей интегрального обслуживания, учитывая множество характеристик узлов коммутации [6].

Развитие многомерных представлений обусловлено необходимостью обработки больших объемов данных с сложной взаимосвязью. Рассмотренные подходы подтверждают актуальность дальнейших исследований в области многомерного хранения, анализа данных, оптимизации систем.

Тензоры и многомерные матрицы. Тензоры широко применяются в математике и физике. Это математический многомерный объект, который задан в векторном пространстве с конечной размерностью и

имеет определенные законы преобразования компонентов. В более широком смысле тензор похож на многомерную матрицу. Многомерная матрица представляет собой набор элементов с набором индексов $i_1, \dots, i_p$, где p – размерность матрицы или ранг для тензора.

Как и с тензорами, с многомерными матрицами можно производить операции: умножения на скаляр, сложение, транспонирование. Но ключевым отличием является произведение матриц.

В алгебре многомерных матриц произведение называется операцией (λ, μ) – свернутым произведением [7].

Для выполнения этой операции у двух многомерных матриц должен быть хотя бы один общий индекс. Набор таких индексов называется набором общих индексов $o = (o_1, \dots, o_\lambda)$. А наборы оставшихся, уникальных для каждой из матриц индексов, называют наборами свободных индексов для двух матриц $r = (r_1, \dots, r_k)$ и $l = (l_1, \dots, l_v)$. λ и μ – это количество скоттовых и кэлиевых индексов произведения. Набор общих индексов разбивается на два набора индексов: скоттовых $s = (s_1, \dots, s_\lambda)$ и кэлиевых $c = (c_1, \dots, c_\mu)$. То есть в зависимости от размера набора общих индексов может быть несколько результатов произведения. Скоттовы индексы остаются в конечном произведении матриц, а по кэлиевым индексам производится сумма и размерность матрицы уменьшается на количество кэлиевых индексов:

$$C = \|c_{rsl}\| = \|\sum_c a_{rsc} \times b_{scl}\|.$$

Здесь $A = \|a_{rsc}\|$ и $B = \|b_{scl}\|$ – наборы элементов двух матриц, а $\|c_{rsl}\|$ результат произведения.

Оптимизация умножения последовательности многомерных матриц. В любой задаче возникает последовательность операций, ведущих к конечному результату. Перемножение матриц – одна из самых затратных операций, а расчет последовательности занимает еще большее время. К тому же выбор последовательности имеет значение, сложность вычисления двух разных последовательностей может кратно различаться. Для двумерных матриц существуют алгоритмы поиска последовательности минимальной сложности. Один из таких методов использует динамическое программирование [8].

В случае многомерных матриц определение оптимальной последовательности уже сложнее. Для двумерного случая известна последовательность и проверить, что эта последовательность возможна, достаточно один раз. В многомерном произведении возможны несколько вариантов произведений матриц в зависимости от количества скоттовых и кэлиевых индексов.

Также возникнет проблема обрывания последовательности. Для примера возьмем набор индексов с размерностями $i1 = 20, i2 = 30, i3 = 60$ и длиной последовательности 4.

Матрицы будут выглядеть следующим образом:

$$A = \{i1\}, B = \{i1, i2, i3\}, C = \{i1, i2, i3\}, D = \{i3\}.$$

Для простоты результатом произведения будет скаляр, то есть каждый индекс будет кэлиевым хотя бы раз в произведении последовательности.

Допустим, первое произведение $B \times C$, где $i2$ и $i3$ являются кэлиевыми индексами, тогда получим следующий набор матриц: $A = \{i1\}, BC = \{i1\}, D = \{i3\}$. Очевидно, что последовательность не может быть продолжена, единственно доступное произведение – это $A \times BC$, и у этого подпроизведения нет общих индексов с D .

Чтобы избежать разрыва связей нужно хранить количество общих индексов. В данном случае имеем (см. таблицы 1 и 2):

Таблица 1

Исходные общие индексы		
$i1$	$i2$	$i3$
3	2	3

Таблица 2

Общие индексы после		
$i1$	$i2$	$i3$
3	0	1

В результате видно, что остается один индекс $i3$ и последовательность произведений невозможна. В случае, когда в результате последовательности ожидается наличие индекса, который являлся кэливым на одном из этапов, таблицу нужно корректировать, вычитая единицу на этапе ее формирования.

При поиске оптимальной последовательности каждая пара формирует несколько вариантов произведений в самом простом случае у пары матриц только один общий индекс, который может быть скоттовым или кэлиевым. В результате имеются два варианта произведения. И с каждым последующим вариантом произведений только больше. Для сокращения таких последовательностей их можно объединять в группы по последовательности записи выражения, так как сложность произведения не меняется в зависимости от порядка перемножения двух матриц: $d(A \times B) = d(B \times A)$, где d как произведение размерностей общих индексов матриц и размерностей свободных индексов.

Данные методы можно применить к поиску оптимальной последовательности. Расчет самой последовательности стоит выполнять параллельно, дальнейший анализ промежуточных результатов алгоритмов может послужить планом параллельного вычисления последовательности.

Заключение. В работе рассмотрены представления данных с использованием тензоров и многомерных матриц, а также задачи оптимизации последовательности их умножения. Проанализированы свойства многомерных матриц и особенности операций над ними, включая свернутое произведение.

Предложенный подход, основанный на учете общих индексов и ограничении допустимых последовательностей, позволяет сократить число вариантов при поиске оптимального решения. Полученные результаты могут быть использованы при разработке эффективных алгоритмов обработки многомерных данных.

Литература

1. Эргашев А.А., Садикова Ф.С. Способы и методы анализа многомерной базы данных // *Universum: технические науки*. 2021. № 12-1 (93). С. 86-89.
2. Смирнов К.Е. Многомерный анализ данных в системах недоопределенных вычислений // *Программные продукты и системы*. 2010. № 4. С. 71-74.
3. Рамазанов З.Р., Катанов Ю.Е. Алгоритм организации многомерных данных при проектировании OLAP кубов // *Прорывные научные исследования как двигатель науки: сборник статей Международной научно-практической конференции*. М. 2021. С. 51.
4. Банин А.А. Применение тензоров в многомерном компонентном анализе категоризованных признаков // *Моделирование и анализ информационных систем*. 2025. Т. 32. № 4. С. 330-359.
5. Сорокин Н. Ф. Применение тензорных сигналов в системах автоматического управления положением мобильных объектов // *Инженерный журнал: наука и инновации*. 2022. № 1 (121). С. 6.
6. Петров М.Н., Колмаков О.В., Иконникова М.Ф. Особенности применения многомерного тензорного анализа при исследовании сетей интегрального обслуживания // *Информатика. Экономика. Управление/Informatics. Economics. Management*. 2023. Т. 2. № 3. С. 101-111.
7. Мунерман В.И. Массовая обработка данных. Алгебраические модели и методы : монография. М.: ИНФРА-М, 2023. 229 с.
8. Ахо А. Построение и анализ вычислительных алгоритмов. М.: Рипол Классик, 1979.

A.S. Mashenkov, V.I. Munerman
Smolensk State University

OPTIMIZATION OF THE SEQUENCE OF MULTIDIMENSIONAL MATRIX MULTIPLICATION

Keywords: *multidimensional data; tensors, multidimensional matrices, contracted product, computational optimization, matrix multiplication order, parallel computing, multidimensional data analysis, data processing.*

Abstract. *The article discusses methods for representing and processing multidimensional data using tensors and multidimensional matrices. The role*

of multidimensional data structures in problems of information analysis and storage is demonstrated, as well as their application in analytics systems and decision-support systems. The properties of multidimensional matrices and operations on them, including the contracted product, are described. The problem of optimizing the sequence of multiplication of multidimensional matrices is considered, the complexity of which significantly depends on the order in which operations are performed. An approach is proposed to reduce the number of admissible sequences by taking into account common indices and preventing breaks in the computational sequence. The possibility of using the obtained results to improve the efficiency of algorithms for multidimensional data processing and for organizing parallel computations is also discussed.

В.Ф. Мирасов, К.А. Антонов, А.А. Назаренко
ООО «НМ-Tech»

УДК 004.032.26

МЕТОДИКА РАСПОЗНАВАНИЯ ДЕФЕКТОВ ИЗДЕЛИЙ МИКРОЭЛЕКТРОНИКИ С ИСПОЛЬЗОВАНИЕМ ПРЕДОБУЧЕННЫХ МОДЕЛЕЙ

Ключевые слова: сверточные нейронные сети, YOLOv12, обучение нейронной сети, детекция дефектов, функции потерь GIoU/CIoU, компьютерное зрение.

В работе рассматривается методика распознавания дефектов изделий микроэлектроники с использованием специализированного программного обеспечения ROVOAM, автоматизирующего процесс подготовки данных и адаптации гиперпараметров под доступные аппаратные ресурсы. Особое внимание уделено архитектуре микросервиса для детекции, алгоритму постобработки результатов и применяемым метрикам оценки качества локализации объектов. Оценка эффективности проведена с использованием метрик mAP и precision–recall, а также показателей производительности. Результаты показывают, что предложенная методика обеспечивает достаточную точность детекции и устойчивость модели при работе в реальном времени.

1. Методика обучения нейросетевой модели с использованием специализированного программного обеспечения ROVOAM. Для решения задачи обучения нейросетевой детекции дефектов разработано специализированное программное обеспечение ROVOAM (Robotic Vision Object Annotation & Modeling). Данный инструмент создан для

автоматизации процесса подготовки и конфигурирования обучающих выборок, что позволяет сократить время развертывания модели и минимизировать ошибки рутинных операций.

Процесс обучения в среде ROVOAM организован следующим образом: на первом этапе пользователем создается новый проект, в рамках которого формируется структура данных. Подготовленный датасет, содержащий размеченные изображения пластин, экспортируется из сторонних инструментов разметки в формате, совместимом с предобученной нейронной сетью YOLO. После импорта единого архивного файла программа автоматически выполняет его распаковку и распределяет изображения по директориям train, valid и test согласно заданным пропорциям.

Ключевым преимуществом разработанного программного обеспечения является адаптивная настройка гиперпараметров обучения в зависимости от доступных аппаратных ресурсов. В условиях ограниченного объема оперативной памяти (8 ГБ) и использования центрального процессора (CPU) для вычислений выбраны консервативные параметры обучения. Размер батча (batch size) установлен равным 2, что позволило избежать переполнения буфера обмена данными. Количество воркеров (workers), отвечающих за параллельную загрузку данных, минимизировано для снижения нагрузки на диск и процессор во время итераций обучения.

Для объективной оценки сходимости модели в процессе тренировки и выявления момента переобучения из общего массива данных выделена тестовая выборка, состоящая из 50 изображений. Данная выборка не участвовала в обновлении весов, а использовалась исключительно для валидации способности нейронной сети к обобщению признаков дефектов.

В ходе экспериментального обучения на собранном датасете модель продемонстрировала высокую эффективность. Визуальный анализ результатов детекции на контрольной выборке показал, что точность обнаружения целевых дефектов составляет от 80 до 90 процентов. Это свидетельствует о корректности выбранной стратегии обучения и достаточной репрезентативности обучающей выборки для решения поставленной задачи автоматического контроля качества изделий микроэлектроники.

2. Архитектура микросервиса и реализация детекции. На базе обученной посредством программного обеспечения ROVOAM модели развернут микросервис в системе автоматизированного контроля качества для решения задачи автоматической детекции объектов на изображениях, поступающих с производственного оборудования. Пользователь загружает изображения через интерфейс системы автоматизированного контроля качества, после чего выполняется их

обработка моделью детекции, формируется таблица результатов и предоставляется возможность просмотра каждой детекции в увеличенном виде при выборе соответствующей строки.

Алгоритмической основой микросервиса детекции дефектов является однократный детектор YOLO (архитектуры v12), реализующий задачу локализации и классификации объектов как единую задачу регрессии. Входное изображение разбивается на сетку $S \times S$, где каждая ячейка предсказывает параметры ограничивающих рамок (bounding boxes), коэффициенты уверенности наличия объекта и вероятности принадлежности к классам. Такой подход обеспечивает высокую скорость работы за счёт однократного прохождения изображения через сеть.

После получения предварительных предсказаний выполняется этап постобработки, включающий фильтрацию результатов и устранение дублирующих ограничивающих рамок методом Non-Maximum Suppression (NMS). Для оценки качества локализации используется метрика Intersection over Union (IoU), определяемая как отношение площади пересечения предсказанной области B и эталонной области B_{gt} к площади их объединения:

$$IoU = \frac{|B \cap B_{gt}|}{|B \cup B_{gt}|}. \quad (1)$$

Данная метрика характеризует степень совпадения ограничивающих рамок и используется как при обучении модели, так и при фильтрации результатов детекции. Однако стандартная метрика IoU не учитывает расстояние между непересекающимися областями, что может замедлять процесс обучения.

Для повышения точности локализации в процессе обучения модели применяются расширенные функции потерь. На этапе дообучения модели в программном обеспечении ROVOAM использовалась функция Generalized Intersection over Union (GIoU), учитывающая площадь минимального ограничивающего прямоугольника C , содержащего обе области:

$$GIoU = IoU - \frac{\left| \frac{C}{B \cup B_{gt}} \right|}{|C|}. \quad (2)$$

Использование данной функции позволяет учитывать пространственное расположение объектов даже при отсутствии пересечения рамок и улучшает сходимость процесса обучения. В современных реализациях YOLO также применяется функция Complete Intersection over Union (CIoU), учитывающая не только перекрытие областей, но и расстояние между центрами рамок и различие их геометрических пропорций:

$$CIoU = IoU - \left(\frac{\rho^2(b, b_{gt})}{c^2} + \alpha v \right), \quad (3)$$

где $\rho(b, b_{gt})$ – евклидово расстояние между центрами предсказанной и эталонной рамок, c – диагональ минимального охватывающего прямоугольника, а величина v характеризует различие отношений сторон ограничивающих областей:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w_{gt}}{h_{gt}} - \arctan \frac{w}{h} \right)^2. \quad (4)$$

Использование CIoU обеспечивает более точную регрессию параметров ограничивающих рамок и повышает устойчивость модели к вариациям размеров объектов.

В рамках разработанного микросервиса обработка изображений выполняется последовательно: входные данные проходят этап предварительной обработки, затем передаются в модель YOLOv12 для предсказания ограничивающих рамок, после чего осуществляется фильтрация результатов и визуализация детекций. Итоговые результаты представляются пользователю в табличной форме, содержащей координаты объектов, вероятности обнаружения и принадлежность к классам, а также предоставляется возможность детального просмотра каждой обнаруженной области. Далее приведен фрагмент изображения с обнаруженным дефектом для детального просмотра (рис. 1).

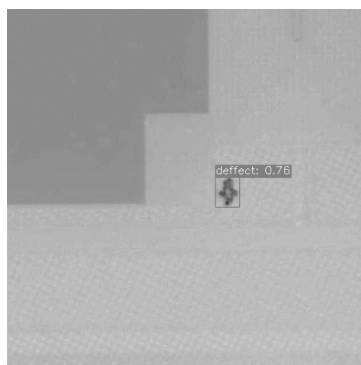


Рис 1. Фрагмент изображения с обнаруженным дефектом

Проведенный анализ метрик детекции, зафиксированных на финальной 74-й эпохе обучения, позволяет сделать вывод об успешной сходимости модели и ее высокой эффективности в решении поставленной задачи.

Модель демонстрирует отличные показатели точности (Precision = 0.988), что свидетельствует о минимальном количестве ложных срабатываний, и высокую полноту (Recall = 0.875), подтверждающую способность системы обнаруживать подавляющее большинство объектов интереса. Значение средней точности mAP50, достигшее 0.976, указывает на надежную классификацию и локализацию объектов при стандартных требованиях к перекрытию.

Более строгая метрика mAP50-95, составившая 0.508, находится на уровне, характерном для современных детекторов, работающих со

сложными мультиклассовыми наборами данных, и подтверждает адекватную точность предсказания ограничивающих рамок.

Сравнение функций потерь на обучающей и валидационной выборках (train/box_loss 1.284 против val/box_loss 1.533) демонстрирует сбалансированность процесса обучения: модель избежала существенного переобучения, сохранив обобщающую способность, необходимую для работы с новыми данными.

Таким образом, экспериментальная оценка системы детекции на базе YOLO v12 подтверждает ее готовность к практическому применению.

Достигнутые показатели точности и полноты обеспечивают высокую надежность работы алгоритма в предполагаемых условиях эксплуатации.

V.F. Mirasov, K.A. Antonov, A.A. Nazarenko
LLC «NM-Tech»

A METHODIC FOR RECOGNITION OF DEFECTS IN MICROELECTRONICS PRODUCTS USING PRE-TRAINED MODELS

Keywords: *convolutional neural networks, YOLOv12, neural network training, defect detection, GIoU/CIoU loss functions, computer vision.*

Abstract. *This paper examines a method for recognizing defects in microelectronics products using specialized ROVOAM software, which automates data preparation and adapts hyperparameters to available hardware resources. Particular attention is paid to the architecture of the detection microservice, the post-processing algorithm, and the metrics used to evaluate the quality of object localization. Efficiency is assessed using mAP and precision-recall metrics, as well as performance indicators. The results demonstrate that the proposed method provides sufficient detection accuracy and model stability in real-time operation.*

В.И. Мунерман, Д.В. Мунерман
Смоленский государственный университет

УДК 004.051

АЛГЕБРАИЧЕСКИЙ ПОДХОД К ПРЕДСТАВЛЕНИЮ СТРУКТУРИРОВАННЫХ ДАННЫХ

Ключевые слова: *модели данных, алгебра кортежей, объектно-ориентированное проектирование и программирование.*

В статье рассматривается одно из основных понятий некоторых, например в широко используемой реляционной модели, моделей данных – понятие кортежа. Поэтому проблема построения абстрактных типов данных (АТД), соответствующих абстрактным алгебраическим машинам, на основе которых строятся такие распространенные модели данных, как реляционная, многомерная, NoSQL, неотделима от проблемы создания алгебры кортежей произвольной структуры. Абстрактная алгебраическая кортеж-машина представляет собой мета-конструкцию, то есть абстрактную структуру, которая формально представляет модель данных независимо от их реального смысла и представления. Рассматривается способ построения бинарных операций над кортежами, без которых невозможно построение бинарных операций над агрегатами данных, которые используются в качестве элементов структуры (отношениями, файлами, многомерными матрицами). В различных моделях данных эти операции над кортежами, заданные явно или неявно, интерпретируются либо как аддитивные, либо как мультипликативные. Предложенный метод позволяет делать явные формальные описания бинарных операций над кортежами.

Введение. Рассматривается одно из основных понятий некоторых, например в широко используемой реляционной модели, моделей данных – понятие кортежа. Поэтому проблема построения абстрактных типов данных (АТД), соответствующих абстрактным алгебраическим машинам, на основе которых строятся такие распространенные модели данных как реляционная, многомерная, NoSQL, неотделима от проблемы создания алгебры кортежей произвольной структуры. Абстрактная алгебраическая кортеж-машина представляет собой мета-конструкцию, то есть абстрактную структуру, которая формально представляет модель данных независимо от их реального смысла и представления. Рассматривается способ построения бинарных операций над кортежами, без которых невозможно построение бинарных операций над агрегатами данных, которые используются в качестве элементов структуры (отношениями, файлами, многомерными матрицами). В различных моделях данных эти операции над кортежами, заданные явно или неявно, интерпретируются либо как аддитивные, либо как мультипликативные. Предложенный метод [1] позволяет делать явные формальные описания бинарных операций над кортежами.

Алгебраическая система кортежей. Далее используется общепринятое в математике определение кортежа как конечного набора $(t_1, \dots, t_n)$ длины n (где n – неотрицательное целое число), каждый элемент которого t_i принадлежит некоторому типу T_i ($1 \leq i \leq n$) [2; 3].

Важную роль в моделях данных играют нуль-кортежи. В дальнейшем нуль-кортеж рассматривается как кортеж, состоящий только из нейтральных элементов типов $T_1, \dots, T_n$.

Предполагается, без ограничения общности, что кортежи не могут содержать сложные элементы, например другие кортежи.

Пусть $T_1, \dots, T_n$ – совокупность основных множеств универсальных алгебр или алгебраических систем, которые принято в языках программирования называть простыми типами. Это, например, числа с фиксированной или плавающей точкой, строки, а также типы, полученные из простых типов добавлением новых операций. Пусть $x_1, \dots, x_p, y_1, \dots, y_q$, ($0 \leq p, q \leq n, p+q=n$) – набор переменных, каждая из которых принимает значения в одном и только одном из множеств $T_1, \dots, T_n$. На этих множествах определяется система функций: $f_{\alpha_1 \dots \alpha_k, \beta_1 \dots \beta_l}^j(x_{\alpha_1}, \dots, x_{\alpha_k}, y_{\beta_1}, \dots, y_{\beta_l}) : (T_{\alpha_1} \times \dots \times T_{\alpha_k} \times T_{\beta_1} \times \dots \times T_{\beta_l}) \rightarrow T_i$. Здесь выполняются неравенства $1 \leq k \leq p$, $1 \leq l \leq q$, $j > 0$ и $1 \leq i \leq n$. Далее будет использоваться сокращенная запись этих функций – $f_{\alpha_1 \dots \alpha_k, \beta_1 \dots \beta_l}^j$.

Операции над кортежами. Пусть кортеж c_1 длины p и кортеж c_2 длины q составлены из переменных $x_1, \dots, x_p$ и $y_1, \dots, y_q$ соответственно. Тогда кортеж c_3 длины r , построенный по правилу $(f_{\alpha_1 \dots \alpha_k, \beta_1 \dots \beta_l}^1, \dots, f_{\alpha_1 \dots \alpha_k, \beta_1 \dots \beta_l}^r)$, можно рассматривать как результат бинарной операции над кортежами c_1 и c_2 ($c_3 = c_1 * c_2$). Если функция определена на всех элементах кортежей c_1 и c_2 , то используется обозначение f_{c_1, c_2}^j .

Семантика операции над кортежами (аддитивность или мультипликативность) определяется семантикой операции, определенной над структурой, типами элементов которой могут быть кортежи c_1, c_2 и c_3 .

В следующем примере показано построение бинарных операций над кортежами.

Пример. Пусть S – множество строк, а R^+ – множество положительных действительных чисел. Переменные (атрибуты) A, B, C, D принимают значения в множестве S , а переменные X, Y – в множестве R^+ . Кортежи c_1 и c_2 имеют схемы $c_1(A, B, C, X)$ и $c_2(B, C, D, Y)$. Функции $f_{c_1, c_2}^2(d) = d, (d \in D)$ позволяют построить кортеж $c_3 = c_1 \times c_2$ со схемой $c_3(A, D, Z)$, где $Z = X \times Y$, и принимают значения в множестве R^+ . Таким образом определяется мультипликативная операция над кортежами. Если c_{31}, c_{32}, c_{33} – кортежи со схемой $c_3(A, D, Z)$, то функции

$$\begin{aligned} f_{c_{31}, c_{32}}^1(c_{31}.a) &= c_{31}.a, (a \in A), \\ f_{c_{31}, c_{32}}^2(c_{31}.d) &= c_{31}.d, (d \in D), \end{aligned}$$

$$f_{c_{31}, c_{32}}^3(c_{31}.z, c_{32}.z) = c_{31}.z \times c_{32}.z, (z \in Z)$$

определяют аддитивную операцию над кортежами $c_{33} = c_{31} + c_{32}$.

Если предположить, что в реляционной модели кортежей c_1 и c_2 есть элементы (строки) отношений R_1 и R_2 , то рассмотренные функции позволяют сформировать список полей в выражении, реализующем запрос:

```
SELECT R1.A, R2.D, Sum(R1.X*R2.Y) AS Z
FROM R1 INNER JOIN R2 ON (R1.C = R2.C) AND (R1.B = R2.B)
GROUP BY R1.A, R2.D.
```

Подобным образом рассмотренные функции могут быть использованы для построения алгебр элементов структур (файлов и многомерных матриц) в теоретико-множественной и многомерно-матричной моделях.

В некоторых моделях при построении бинарных операций над элементами структуры может возникнуть ситуация, когда кортеж-результат формируется только из одного кортежа-операнда. В многомерно-матричной модели такое невозможно, так как в матрице присутствуют все возможные элементы, в том числе и нуль-кортежи. Но в реляционной и теоретико-множественной моделях отношения и файлы, как правило, не содержат строки и записи, состоящие только из нейтральных элементов. Поэтому целесообразно рассмотреть еще два вида функций $f_{\alpha_1 \dots \alpha_k, 0}^j(x_{\alpha_1}, \dots, x_{\alpha_k}) : (T_{\alpha_1} \times \dots \times T_{\alpha_k}) \rightarrow T_j$ и $f_{0, \beta_1 \dots \beta_l}^j(y_{\beta_1}, \dots, y_{\beta_l}) : (T_{\beta_1} \times \dots \times T_{\beta_l}) \rightarrow T_j$, которые позволят конструировать операции, формирующие кортеж-результат только из одного кортежа-операнда. Использование этих операций обеспечивает единство формальной записи алгоритмов бинарных операций над структурами. Сокращенная запись этих функций имеет вид: $f_{0, \beta_1 \dots \beta_l}^j, f_{\alpha_1 \dots \alpha_k, 0}^j$.

Предложенный способ позволяет определять различные аддитивные и мультипликативные операции над кортежами и получать при этом различные универсальные алгебры или алгебраические системы. Свойства построенных операций составляют список аксиом этих алгебраических систем. Благодаря этому возникает возможность формального построения произвольных универсальных алгебраических систем кортежей в соответствии с требованиями реальных задач.

Алгебра кортежей и объектно-ориентированный подход. От построенной таким образом универсальной алгебраической системы легко перейти к объектному [4; 5] представлению АТД *Кортеж*. Такой АТД является базовым объектом, который может стать родоначальником множества объектов, соответствующих реальным задачам. Эти объекты будут, с одной стороны, наследовать свойства составляющих кортежи простых типов, с другой, – базовые операции

объекта, представляющего АТД *Кортеж*. Таким образом, в объектах-наследниках могут использоваться либо основные операции над кортежами из базового объекта, либо заменяющие их полиморфные операции, которые будут выполнять те действия над элементами простых типов данных, которые требуются в условиях конкретной решаемой задачи.

Заключение. Предложенный способ построения универсальной алгебраической системы для кортежей может быть использован в дальнейшем для построения второго основного множества (типа) в абстрактных файловой и матричной алгебраических машинах. Он также будет использован для доказательства изоморфизма различных моделей данных и вычислений.

Литература

1. Мунерман В.И. Массовая обработка данных. Алгебраические модели и методы: монография. М.: ИНФРА-М, 2023. 229 с.
2. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. Киев: Наукова думка, 1989. 376 с.
3. Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002. 800 с.
4. Liskov B., Zilles S. Programming with abstract data types // ACM SIGPLAN Notices Volume 9 Issue 4, April 1974. P. 50–59.
5. Liskov B. et al. Abstraction mechanisms in CLU // Communications of the ACM. 1977. T. 20, №. 8. С. 564-576.

V.I. Munerman, D.V. Munerman
Smolensk State University

ALGEBRAIC APPROACH TO STRUCTURED DATA REPRESENTING

Keywords: *data models, tuple algebra, object-oriented design and programming.*

Abstract. *This article examines one of the fundamental concepts of some data models, such as the widely used relational model – the concept of a tuple. Therefore, the problem of constructing abstract data types (ADTs) corresponding to abstract algebraic machines, which underlie such common data models as relational, multidimensional, and NoSQL, is inseparable from the problem of creating a tuple algebra of arbitrary structure. An abstract algebraic tuple machine is a meta-construct, that is, an abstract structure that formally represents a data model, regardless of their actual meaning and representation. This article discusses a method for constructing binary operations on tuples, which are essential for constructing binary operations on data aggregates used as structure elements (relations, files, multidimensional matrices). In various data models, these operations on*

tuples, whether specified explicitly or implicitly, are interpreted as either additive or multiplicative. The proposed method enables explicit formal descriptions of binary operations on tuples

В.Д. Никаноров, Ю.С. Шевнина, Е.Л. Федотова
Национальный исследовательский университет «МИЭТ»

УДК 517.977.58

ПРЕДИКТИВНОЕ УПРАВЛЕНИЕ НЕЛИНЕЙНОЙ СИСТЕМОЙ НА ОСНОВЕ ОПЕРАТОРА КУПМАНА

Ключевые слова: нелинейные динамические системы, оператор Купмана, extended dynamic mode decomposition with control (EDMDc), управление нелинейными системами, модельно-предиктивное управление (MPC), слежение за траекторией.

В статье рассматривается задача управления нелинейной динамической системой при отсутствии аналитического описания её математической модели. Предложен подход, основанный на аппроксимации оператора Купмана методом EDMDc и применении линейного предиктивного управления (MPC) в пространстве наблюдаемых функций. Выполнено численное моделирование на примере нелинейной системы маятника при решении задачи слежения за траекторией. Проведен анализ эффективности подхода, а также исследовано влияние объёма обучающей выборки на качество управления.

1. Постановка задачи. Рассматривается дискретная по времени нелинейная динамическая система

$$x_{k+1} = f(x_k, u_k), \quad x_k \in \mathbb{R}^n, \quad u_k \in \mathcal{U} \subset \mathbb{R}^m,$$

где $k=0,1,2,\dots$ – дискретные моменты времени, x_k – состояние, u_k – управление, $\mathcal{U}$ – множество допустимых управлений, а функция $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ описывает динамику системы, при этом ее аналитическое выражение неизвестно.

Предполагается, что задан экспериментальный набор данных, полученный при функционировании системы:

$$\mathcal{D} = \{(x_k, u_k, x_{k+1})\}_{k=0}^{M-1},$$

где M – объём обучающей выборки. Вектор состояния x_k считается измеримым.

Пусть задана эталонная траектория $\{r_k\}_{k \geq 0}$, $r_k \in \mathbb{R}^n$ и горизонт планирования $T \in \mathbb{N}$. Качество управления оценивается целевой функцией

$$J_k = \sum_{i=0}^{T-1} \left(\|x_{k+i} - r_{k+i}\|_Q^2 + \|u_{k+i}\|_R^2 \right),$$

где $Q \succeq 0$, $R \succ 0$. Взвешенная норма определяется как $\|x\|_Q^2 = x^\top Q x$.

Требуется по данным $\mathcal{D}$ на каждом шаге k решить задачу оптимизации для нахождения последовательности управлений

$$\bar{u}_k^* = \arg \min_{\bar{u}_k \in U^T} J_k(x_k, \bar{u}_k),$$

где $\bar{u}_k = (u_k, \dots, u_{k+T-1})^\top$ – вектор управлений на горизонте прогнозирования.

2. Аппроксимация оператора Купмана. Рассмотрим дискретную управляемую систему $x_{k+1} = f(x_k, u_k)$. Обозначим $\mathcal{X} \subseteq \mathbb{R}^n$ область допустимых состояний системы. Оператор Купмана [1] K определяется как

$$(Kg)(x, u) = g(f(x, u))$$

и действует на пространстве наблюдаемых функций $g: \mathcal{X} \rightarrow \mathbb{R}$. Он позволяет представить нелинейную динамику как линейную в бесконечномерном пространстве наблюдаемых функций. В практических задачах используется конечномерная аппроксимация оператора Купмана, что позволяет применять к нелинейной системе линейные методы управления в расширенном пространстве наблюдаемых функций [4]. В данной работе для аппроксимации оператора Купмана используется метод extended dynamic mode decomposition with control (EDMDc) [3].

Пусть задан набор выбранных наблюдаемых функций $\psi: \mathbb{R}^n \rightarrow \mathbb{R}^p$ и соответствующие переменные $z_k = \psi(x_k)$. Динамика состояния z_k аппроксимируется линейной системой $z_{k+1} \approx Kz_k + Bu_k$, $K \in \mathbb{R}^{p \times p}$, $B \in \mathbb{R}^{p \times m}$.

По имеющимся данным $\mathcal{D} = \{(x_k, u_k, x_{k+1})\}_{k=0}^{M-1}$ формируются матрицы $\Psi_X = [\psi(x_0) \dots \psi(x_{M-1})]$, $\Psi_Y = [\psi(x_1) \dots \psi(x_M)]$, $U = [u_0 \dots u_{M-1}]$.

Матрицы K и B находятся как решение задачи наименьших квадратов в норме Фробениуса $\|\cdot\|_F$.

$$\min_{K, B} \|\Psi_Y - K\Psi_X - BU\|_F^2.$$

Для регуляризации решения вводится объединённая матрица $\Theta = [K \ B]$ и расширенный регрессор $W = \begin{bmatrix} \Psi_X \\ U \end{bmatrix}$, после чего искомая матрица вычисляется по формуле $\Theta = \Psi_Y W^\top (WW^\top + \lambda I)^{-1}$, где $\lambda \geq 0$ – параметр регуляризации, I – единичная матрица размерности $(p+m) \times (p+m)$.

Восстановление состояния x_k в исходном пространстве по наблюдаемым переменным z_k осуществляется с помощью линейного оператора $C \in \mathbb{R}^{n \times p}$, полученного из регуляризованной задачи регрессии

$$\min_C \|X - C\Psi_X\|_F^2 + \mu \|C\|_F^2, \quad C = X\Psi_X^\top (\Psi_X\Psi_X^\top + \mu I)^{-1},$$

где $X=[x_0 \dots x_{M-1}]$, а $\mu \geq 0$ – параметр регуляризации.

3. Линейное предиктивное управление в расширенном пространстве. После получения аппроксимации линейного оператора Купмана становится возможным использовать для управления нелинейной системой методы линейного управления, применяемые в пространстве наблюдаемых функций [4]. В данной работе для этого используется линейное предиктивное управление (МРС) [4; 5]:

$$z_{k+1} = Kz_k + Bu_k, \quad x_k = Cz_k,$$

где $z_k \in \mathbb{R}^p$ – вектор наблюдаемых переменных, $u_k \in \mathbb{R}^m$ – управление, $x_k \in \mathbb{R}^n$ – состояние в исходном пространстве, матрицы K , B , C получены методом EDMDc (раздел 2).

Для горизонта прогнозирования T введём предсказательные матрицы $\mathcal{P}$ и $\mathcal{H}$ такие, что

$$Z = \mathcal{P}z_k + \mathcal{H}U, \quad X = (I_T \otimes C)Z = \mathcal{P}_x z_k + \mathcal{H}_x U,$$

где $U = (u_k, \dots, u_{k+T-1})^\top$ – вектор управлений, $Z = (z_{k+1}, \dots, z_{k+T})^\top$ – вектор прогнозируемых наблюдаемых, $X = (x_{k+1}, \dots, x_{k+T})^\top$ – прогноз состояний в исходном пространстве, $\mathcal{P}_x = (I_T \otimes C)\mathcal{P}$, $\mathcal{H}_x = (I_T \otimes C)\mathcal{H}$.

Матрицы $\mathcal{P}$ и $\mathcal{H}$ имеют стандартную для МРС структуру:

$$\mathcal{P} = \begin{bmatrix} K \\ K^2 \\ \vdots \\ K^T \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} B & 0 & \dots & 0 \\ KB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ K^{T-1}B & K^{T-2}B & \dots & B \end{bmatrix}.$$

Целевая функция определяется как $J_k = (X - R_{\text{ref}})^\top \bar{Q}(X - R_{\text{ref}}) + U^\top \bar{R}U$,

где $\bar{Q} = \text{diag}(Q, \dots, Q, P)$ и $\bar{R} = \text{diag}(R, \dots, R)$ – блочно-диагональные весовые матрицы, $Q \succeq 0$, $R \succ 0$, $P \succeq 0$, а $R_{\text{ref}} = (r_{k+1}, \dots, r_{k+T})^\top$ – вектор эталонной траектории.

Подстановка линейной модели $X = \mathcal{P}_x z_k + \mathcal{H}_x U$ приводит к выпуклой задаче квадратичного программирования:

$$\min_U \frac{1}{2} U^\top \mathcal{H}U + f^\top U \quad \text{при} \quad GU \leq h,$$

где $\mathcal{H}_{qp} = 2(\mathcal{H}_x^\top \bar{Q} \mathcal{H}_x + \bar{R})$, $f = 2\mathcal{H}_x^\top \bar{Q}(\mathcal{P}_x z_k - R_{\text{ref}})$, а матрицы G , h задают покомпонентные ограничения на управление:

$$u_{\min} \leq (U)_j \leq u_{\max}, \quad j = 1, \dots, Tm.$$

На каждом шаге решается сформулированная QP-задача, после чего к объекту прикладывается первое управление u_k из найденной последовательности. Для эффективной реализации матрицы $\mathcal{P}_x, \mathcal{H}_x, \mathcal{H}_{qp}, G, h$ вычисляются однократно до начала управления, а в реальном времени пересчитывается лишь линейный член f , зависящий от текущих состояния z_k и сегмента эталонной траектории.

4. Численный эксперимент. Для оценки эффективности полученных результатов проведен численный эксперимент на модели маятника $\ddot{\theta} = -\frac{g}{L} \sin \theta - \frac{b}{m} \dot{\theta} + \frac{1}{mL^2} u$.

Обучающая выборка $\mathcal{D} = \{(x_k, u_k, x_{k+1})\}_{k=0}^{M-1}$ состояла из состояний и управлений, покрывающих широкий диапазон углов $[-\pi, \pi]$ и скоростей $[-3, 3]$ рад/с, что обеспечивает репрезентативность данных. Оператор Купмана аппроксимировался методом EDMDc с полиномами до третьей степени включительно в качестве наблюдаемых функций.

Цель управления – слежение за периодической эталонной траекторией длиной 200 шагов, построенной так, что угол изменяется в пределах $[0, \pi]$. Параметры MPC: горизонт планирования $T = 25$, весовые матрицы $Q = \text{diag}(100, 0)$, $R = 0.005$, ограничения: $|u| \leq 100 \text{ Н} \cdot \text{м}$.

Качество управления оценивалось интегральной стоимостью

$$J = \sum_{k=0}^{T-1} (\|x_k - r_k\|_Q^2 + \|u_k\|_R^2) + \|x_T - r_T\|_P^2.$$

В качестве эталона использовалось оптимальное управление, полученное минимизацией того же функционала по точной нелинейной модели. Результаты, полученные при объеме обучающей выборки $M = 80$, представлены на рисунке 1, при этом полученная интегральная стоимость превысила оптимальную на 4.67%.

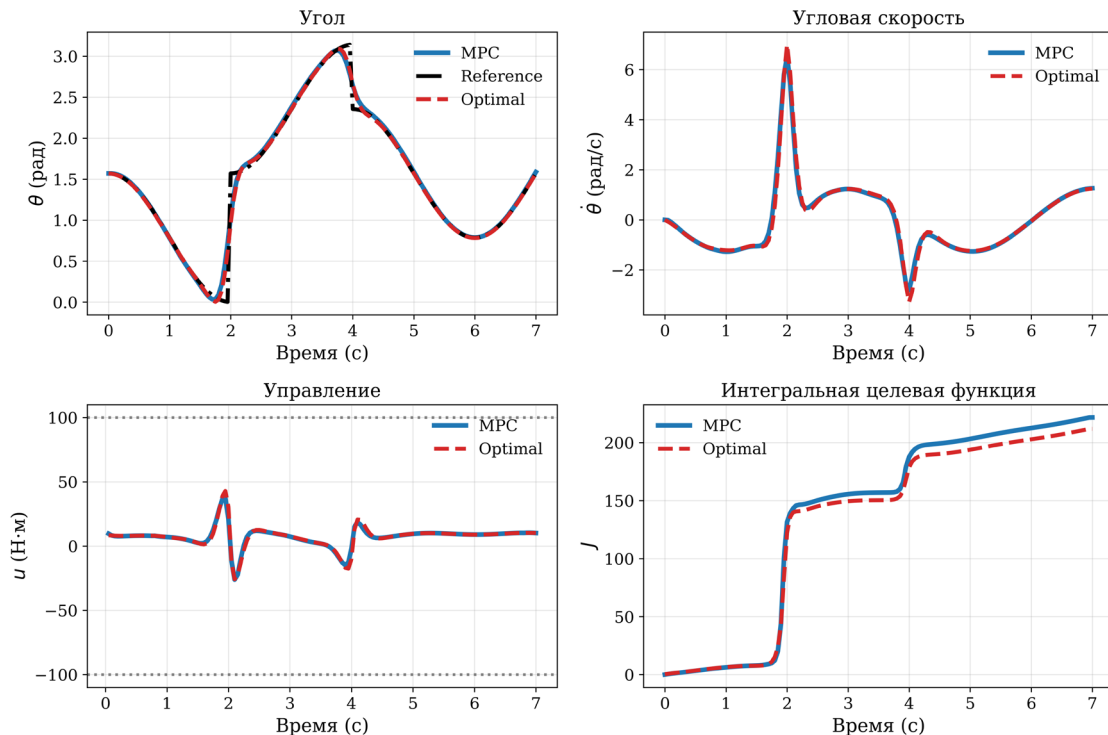


Рис. 1. Управление системой маятника при объеме обучающей выборки $M = 80$.

В процессе исследования проведен анализ зависимости качества управления от размера обучающей выборки, результаты представлены в таблице 1.

Таблица 1

Зависимость качества управления от размера обучающей выборки

Размер обучающей выборки M	80	160	320	800	5000
Интегральная стоимость J	221.6	216.0	213.5	212.5	212.2
Превышение стоимости оптимального управления, %	4.67	2.05	0.87	0.39	0.24

Рассмотренный подход позволяет эффективно управлять нелинейной системой без знания ее математической модели и при малом объеме данных. Основным фактором, определяющим точность, является репрезентативность обучающей выборки, что согласуется с теорией аппроксимации оператора Купмана [3].

Литература

1. Koopman B.O. Hamiltonian systems and transformation in Hilbert space // Proceedings of the National Academy of Sciences. 1931. Vol. 17, No. 5. P. 315–318.
2. Proctor J.L., Brunton S.L., Kutz J.N. Dynamic mode decomposition with control // SIAM Journal on Applied Dynamical Systems. 2016. Vol. 15, No. 1. P. 142–161.
3. Williams M.O., Kevrekidis I.G., Rowley C.W. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition // Journal of Nonlinear Science. 2015. Vol. 25, No. 6. P. 1307–1346.
4. Korda M., Mezić I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control // Automatica. 2018. Vol. 93. P. 149–160.
5. Rawlings J.B., Mayne D.Q., Diehl M. Model Predictive Control: Theory, Computation, and Design. Madison: Nob Hill Publishing, 2017. 770 p.

V.D. Nikanorov, Yu.S. Shevnina, E.L. Fedotova

National Research University of Electronic Technology (MIET)

KOOPMAN-BASED MODEL PREDICTIVE CONTROL FOR NONLINEAR SYSTEMS

Keywords: *nonlinear dynamical systems, Koopman operator, extended dynamic mode decomposition with control (EDMDc), control of nonlinear systems, model predictive control (MPC), trajectory tracking.*

Abstract. *This paper addresses the control problem for a nonlinear dynamical system without an analytical description of its mathematical model. A data-driven approach is proposed, combining the Koopman operator approximation via the EDMDc method with linear model predictive control (MPC) in the lifted space of observables. Numerical simulations are carried out for a nonlinear pendulum system in a trajectory tracking scenario. The effectiveness of the proposed approach is evaluated, and the impact of the training dataset size on control quality is analyzed.*

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СЕТЕЙ СВЯЗИ В УСЛОВИЯХ МАСШТАБИРУЕМОСТИ

Ключевые слова: *сети связи, масштабируемость, расширяемость графов, паросочетания, компьютерное моделирование, топология сетей.*

В статье рассматривается задача математического моделирования процессов функционирования сетей связи при масштабировании топологии. В отличие от классических подходов, ориентированных на спектральные характеристики расширяющих графов, предлагается использовать теоретико-графовое свойство расширяемости как меру способности сети сохранять связность и отказоустойчивость при наращивании числа узлов. Установлена связь между расширяемостью графа и его способностью поддерживать паросочетания при масштабировании. Получены оценки сложности процессов маршрутизации в терминах расширяемости.

1. Введение. Современные сети связи характеризуются постоянным ростом числа узлов и объема трафика, что делает проблему масштабируемости одной из ключевых при проектировании сетевых архитектур. Под масштабируемостью понимается способность сети сохранять заданные показатели производительности (пропускную способность, задержки, отказоустойчивость) при увеличении количества абонентских устройств и коммуникационного оборудования [1; 2].

Традиционно для моделирования топологий масштабируемых сетей используются расширяющие графы [3]. Это разреженные графы с высокими показателями связности, измеряемыми через спектральный зазор лапласиана. Такие графы лежат в основе топологий Jellyfish, Xpander, STRAT, применяемых в центрах обработки данных. Однако спектральный подход имеет ограничения: он характеризует связность графа в целом, но не дает прямой информации о том, как именно можно наращивать сеть, сохраняя ее структурные свойства [3; 4].

В настоящей работе предлагается иной подход, основанный на комбинаторном свойстве расширяемости графа. Это свойство, введенное Пламмером и развитое в теории паросочетаний, характеризует способность любого заданного множества независимых ребер быть дополненным до совершенного паросочетания [5; 6]. В контексте сетей [8] связи это интерпретируется следующим образом: если в

функционирующей сети выделен набор активных соединений (паросочетание), то расширяемость гарантирует возможность установления дополнительных соединений без разрыва существующих.

Целью работы является построение и обоснование математической модели процессов функционирования сетей связи в условиях масштабирования, где свойство расширяемости выступает как количественная мера структурной гибкости сети.

2. Теоретические основы расширяемости графов. Следуя классическому определению [3; 6; 7], граф G с совершенным паросочетанием называется n -расширяемым, если:

- 1) G имеет паросочетание размера n ;
- 2) любое паросочетание размера n в G может быть расширено до совершенного паросочетания.

Числом расширяемости $ext(G)$ называется наибольшее целое n , для которого граф является n -расширяемым. Для существования n -расширяемости необходимо выполнение условия:

$$1 \leq n \leq (p/2) - 1, \quad (1)$$

где $p = |V(G)|$.

Для двудольных графов с биразбиением (U, V) существует эффективный алгоритм вычисления расширяемости, основанный на анализе остаточного графа G_M , получаемого ориентацией ребер паросочетания M . Граф G является k -расширяемым тогда и только тогда, когда его остаточный граф сильно связан и между каждой вершиной из U и каждой вершиной из V существует k вершинно-непересекающихся ориентированных путей.

Для моделирования сетей связи введем следующие интерпретации:

1. Вершины графа – узлы сети (коммутаторы, маршрутизаторы, абонентские устройства).
2. Ребра графа – каналы связи между узлами.
3. Паросочетание – множество непересекающихся по вершинам активных соединений (например, виртуальных каналов или туннелей), что исключает конфликты при коммутации.
4. Совершенное паросочетание – состояние сети, в котором задействованы все узлы (предельная загрузка).
5. Масштабирование – процедура добавления новых вершин и ребер к существующему графу.

Свойство расширяемости приобретает следующий смысл: если сеть находится в состоянии с n активными непересекающимися соединениями, то n -расширяемость гарантирует возможность перехода к режиму полной загрузки без переконфигурации существующих соединений.

Пусть сеть связи моделируется графом G с $ext(G) \geq n$. Тогда для любого множества из $m \leq n$ не пересекающихся по узлам активных соединений существует алгоритм достройки сети до состояния, в котором задействованы все узлы, не требующий разрыва существующих соединений. Непосредственно следует из определения n -расширяемости: любое паросочетание размера $m \leq n$ является подмножеством некоторого паросочетания размера n , а любое паросочетание размера n расширяется до совершенного. Транзитивность расширения гарантирует существование совершенного паросочетания, содержащего исходное m -элементное множество ребер.

3. Моделирование процессов масштабирования. Рассмотрим процесс масштабирования сети как последовательность шагов, на каждом из которых к графу $G_t = (V_t, E_t)$ добавляется новый узел v_{new} и соединяется с существующими узлами. Требуется, чтобы в процессе масштабирования сеть сохраняла способность к расширению паросочетаний.

Определение 1. Масштабируемым расширяющим семейством назовем последовательность графов $\{G_t\}_{t=1}^{\infty}$ такую, что:

- 1) G_{t+1} получается из G_t добавлением одной вершины и нескольких ребер;
- 2) $\forall t : ext(G_{t+1}) > ext(G_t)$;
- 3) $\exists c > 0 : ext(G_t) \geq c \cdot |V_t| / 2$.

Условие (2) гарантирует, что с ростом сети ее расширяемость не ухудшается, условие (3) обеспечивает линейный рост числа гарантированно расширяемых паросочетаний относительно размера сети.

Для класса регулярных графов, представляющих особый интерес при проектировании сетей связи (топологии типа «каждый узел имеет одинаковое число портов»), справедлива следующая оценка.

Теорема 1. Для бесконечного семейства k -регулярных графов $\{G_t\}$ с числом вершин $p_t \rightarrow \infty$ и $ext(G_t) \geq [k/2]$ существует алгоритм масштабирования, при котором добавление новой вершины требует не более k новых ребер и сохраняет расширяемость.

Доказательство. Конструктивно. Пусть G_t удовлетворяет условию. Выберем в G_t паросочетание M_t размера $ext(G_t)$. Добавим вершину v_{new} и соединим ее с k вершинами графа G_t так, чтобы:

- среди выбранных вершин не более одной, инцидентной каждому ребру из M_t ;
- новые ребра не создавали кратных.

Построенный граф G_{t+1} имеет $p_t + 1$ вершин и является k -регулярным. Для доказательства сохранения расширяемости рассмотрим произвольное паросочетание N размера $ext(G_t)$ в G_{t+1} . Если $v_{new} \notin V(N)$, то N является паросочетанием в G_t и расширяется до совершенного паросочетания в G_t , которое автоматически является совершенным в G_{t+1} . Если v_{new} инцидентна ребру из N , то используем технику чередующихся цепей для замены этого ребра с сохранением расширяемости. ■

4. Моделирование процессов маршрутизации. Рассмотрим процесс установления соединений в масштабируемой сети. Пусть в момент времени t сеть описывается графом G_t с расширяемостью $ext(G_t) = n_t$. Поступает запрос на установление m новых непересекающихся соединений.

Сформируем алгоритм динамического расширения паросочетаний:

1. Имеется текущее множество активных соединений M_{curr} (паросочетание).
2. Поступает множество запросов RR на новые соединения, не конфликтующие с M_{curr} .
3. Если $|M_{curr}| + |R| \leq n_t$, то применяем теорему о расширении паросочетаний: существует совершенное паросочетание, содержащее $M_{curr} \cup R$.
4. Находим такое совершенное паросочетание с помощью алгоритма чередующихся цепей (сложность $O(|E|)$).
5. Устанавливаем новые соединения из R , сохраняя существующие из M_{curr} .

Теорема 2. Алгоритм 1 корректен и требует не более $O(|V_t| \cdot |E_t|)$ операций на шаг.

Доказательство. Корректность следует из свойства nt -расширяемости. Оценка сложности вытекает из того, что поиск расширяющей цепи в графе с паросочетанием выполняется за линейное время, а в худшем случае требуется $O(|V_t|)$ таких поисков. ■

Предложенный подход развивает идеи, заложенные в работах по расширяющим графам, но смещает акцент со спектральных характеристик на комбинаторные. Ближайшим аналогом является понятие жесткости графов и d -мерной алгебраической связности. Однако если жесткость характеризует способность графа сохранять структуру при деформациях вложения в пространство, то расширяемость описывает комбинаторную гибкость при перестройке паросочетаний.

Для приложений в проектировании сетей связи важны оба подхода:

- спектральные характеристики важны для равномерного распределения нагрузки и малых диаметров;
- расширяемость важна для гарантированного установления соединений без конфликтов.

Перспективным направлением представляется построение графов, одновременно обладающих хорошими спектральными свойствами и высокой расширяемостью. Такие графы могут стать основой для следующего поколения масштабируемых сетевых архитектур.

5. Заключение. В работе предложена математическая модель процессов функционирования сетей связи в условиях масштабируемости, основанная на свойстве расширяемости графов; установлена интерпретация расширяемости как меры способности сети к наращиванию числа соединений без конфликтов; предложен алгоритм динамического управления соединениями, использующий свойство расширяемости.

Дальнейшие исследования предполагают анализ сложности поддержания расширяемости в динамических условиях, а также построение явных конструкций графов, сочетающих высокую расширяемость с малым диаметром и спектральными свойствами расширителей.

Литература

1. О связи: Федеральный закон от 7 июля 2003 г. № 126-ФЗ. URL: https://www.consultant.ru/document/cons_doc_LAW_43224/ (дата обращения 12.02.2026).
2. Об информации, информационных технологиях и о защите информации : Федеральный закон № 149-ФЗ. URL: https://www.consultant.ru/document/cons_doc_LAW_61798/ (дата обращения 12.02.2026).
3. Акимов О. Е. Дискретная математика: логика, группы, графы. 2-е изд., доп. М.: Лаборатория базовых знаний, 2003. 376 с.
4. Зыков А.А. Основы теории графов. М.: Вузовская книга, 2004. 662 с.
5. Plummer M.D. On n -extendible graphs // Discrete Mathematics. 1980. Vol. 31, No. 2. P. 201–210.
6. Kolenbrander J. Expander Graphs, Lifts, and Applications. Princeton University Senior Theses, 2024.
7. Lew A., Nevo E., Peled Y., Raz O.E. Rigidity expander graphs // Combinatorica. 2025. Vol. 45. Article 24.
8. Melnikov B., Terentyeva Y. Разработка больших сетей связи // Современные информационные технологии и ИТ-образование. 2021. Т. 17, № 1. С. 69–79.

A.V. Popov

National Research University of Electronic Technology (MIET)

MODELING COMMUNICATION NETWORK OPERATIONAL PROCESSES UNDER SCALABILITY

Keywords: *communication networks, scalability, graph extensibility, matchings, computer modeling, network topology.*

Abstract. *This paper examines the problem of mathematical modeling of communication network operations under topology scaling. Unlike classical approaches focused on the spectral characteristics of expanding graphs, it is proposed to use the graph-theoretic property of extensibility as a measure of a network's ability to maintain connectivity and fault tolerance as the number of nodes increases. A relationship is established between graph extensibility and its ability to maintain matchings under scaling. Estimates of the complexity of routing processes in terms of extensibility are obtained.*

Д.С. Редченков, Т.А. Самойлова
Смоленский государственный университет

УДК 004.032.24.

АНАЛИЗ МОДЕЛЕЙ ПАРАЛЛЕЛИЗМА В СИСТЕМАХ ВЗАИМОДЕЙСТВИЯ КЛИЕНТ-СЕРВЕРА

Ключевые слова: *параллелизм, REST API, Go, FastAPI, ASP.NET Core, k6, производительность, латентность, высоконагруженные сервисы.*

В статье проводится сравнительный анализ трёх подходов к параллельной обработке HTTP-запросов: потоковой модели Go net/http, событийной модели Python/FastAPI и модели пула потоков ASP.NET Core Minimal API. Все три сервиса построены на едином API-контракте, одинаковой in-теторгу логике и идентичном наборе данных. Нагрузка создавалась инструментом k6 в двух режимах: с преобладанием операций ввода-вывода и с преобладанием вычислений. Эксперимент показал, что Go и ASP.NET демонстрируют сопоставимые результаты в обоих режимах, тогда как FastAPI заметно уступает при интенсивных вычислениях. По итогам исследований сформулированы рекомендации по выбору серверной платформы под конкретный профиль нагрузки.

1. Введение. Выбор серверного стека для REST API при высокой конкурентной нагрузке остаётся практически значимой задачей, где такое решение нередко принимается интуитивно, опираясь на опыт конкретной команды или на синтетические тесты. Большинство известных сравнений проводятся в неравных условиях: различный API-контракт, разная логика, неодинаковые данные и параметры нагрузки не

позволяют делать обоснованные выводы. Между тем модель параллелизма во время выполнения программы напрямую влияет на стоимость переключения контекста, планирование задач и деградацию под нагрузкой.

В статье рассматриваются Go net/http [1], Python/FastAPI (ASGI/asyncio) [2; 3] и ASP.NET Core (Kestrel + thread pool) [4]. Цель работы состоит в воспроизводимом сравнении этих трёх платформ на одной задаче REST API при минимальном влиянии внешних факторов. Для этого потребовалось привести контракт и поведение всех трёх сервисов к единому виду, унифицировать формат ошибок и трассировку запросов, сформировать сценарии нагрузки с явным разделением на I/O и CPU-профили, а затем провести повторяемые прогоны и проанализировать медианные метрики. Итогом стали практические рекомендации по выбору платформы под конкретный профиль нагрузки.

2. Дизайн эксперимента. Сравняются три сервиса: Go (gonethttp), Python (py-fastapi) и ASP.NET (cs-webapi). Все реализуют одинаковые эндпоинты: GET /ping, GET /api/user/{id}, POST /api/user, GET /cpu-like?n=200000. Предметная логика работает in-memory без обращений к базе данных, набор данных идентичен (id = 1 ... 10000), схема ответов и политика X-Request-Id унифицированы. Нагрузка создавалась четырьмя сценариями k6 [5]: hello.js оценивает базовые накладные расходы HTTP; routes_param.js проверяет чтение по параметризованному маршруту; json_small.js включает смешанные операции с сериализацией JSON; cpu_like.js запускает одинаковый вычислительный цикл во всех трёх реализациях. Фиксировались пропускная способность (RPS), перцентили задержек p50, p90, p95, p99 и доля ошибок.

3. Результаты эксперимента. В I/O сценариях (рис. 1) все три сервиса показывают близкие результаты в hello, что подтверждает корректность стенда. При появлении бизнес-логики и сериализации Go и ASP.NET начинают опережать FastAPI: в routes_param разрыв составляет около 33%, в json_small достигает 53%.

В сценарии CPU (рис. 2) картина меняется кардинально: Go показывает 376,99 RPS, ASP.NET – 374,26 RPS, FastAPI – 9,39 RPS. Разрыв между компилируемыми языками и Python составляет около 40 раз. Причина в том, что сценарий почти целиком состоит из вычислений на уровне Python-кода: стоимость одной итерации здесь несравнимо выше, чем в Go или .NET. При росте числа одновременных запросов очередь накапливается быстрее, чем успевает разгружаться, и RPS резко падает.

Анализ задержек в CPU (рис. 3) наглядно отражает разрыв в три порядка: FastAPI p50 = 1978 мс, p90 = 2876 мс, p99 = 3413 мс, тогда как у Go p50 = 1,95 мс, p90 = 3,04 мс, p99 = 4,99 мс, у ASP.NET p50 = 2,17 мс,

p90 = 3,48 мс, p99 = 5,90 мс. Доля ошибок во всех прогонах равна 0,00%, что подтверждает функциональную корректность всех трёх сервисов.

Сравнение медианных p95 по всем сценариям (рис. 4) подтверждает, что отставание FastAPI заметно не только в `cpu_like`, но и в `routes_param` и `json_small`.

Коэффициент вариации RPS по трём прогонам (рис. 5) у Go и ASP.NET не превышает 0,17% ни в одном сценарии. У FastAPI в I/O сценариях он достигает 4,88%, однако это объясняется нестабильностью очереди при высокой нагрузке, а не погрешностью измерений. В `cpu_like`, где FastAPI работает далеко от предела насыщения, CV составляет всего 0,32%.

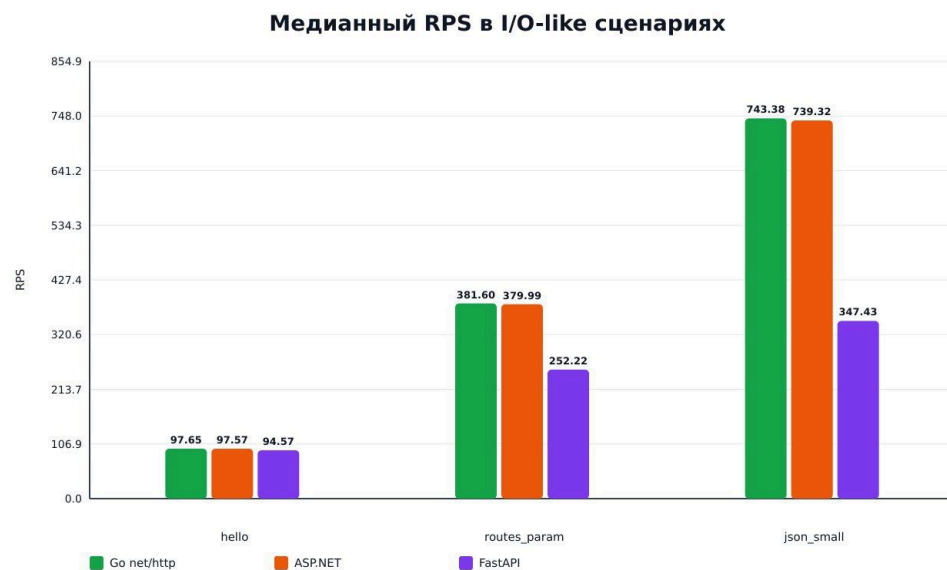


Рис. 1. Медианный RPS в I/O сценариях

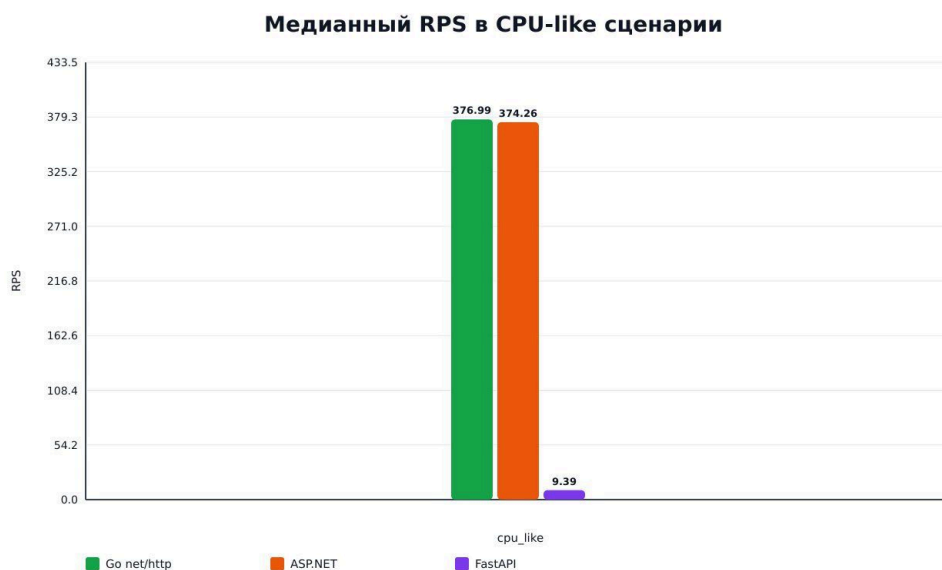


Рис. 2. Медианный RPS в CPU-сценарии

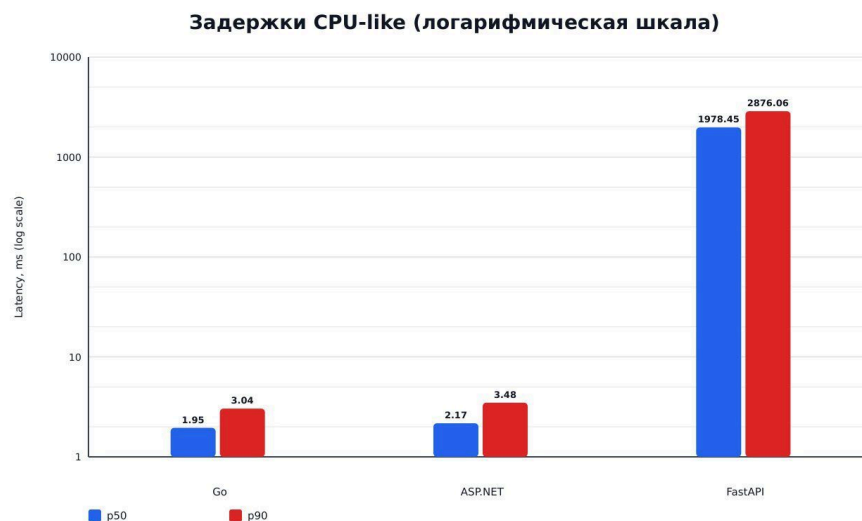


Рис. 3. Задержки CPU (логарифмическая шкала)

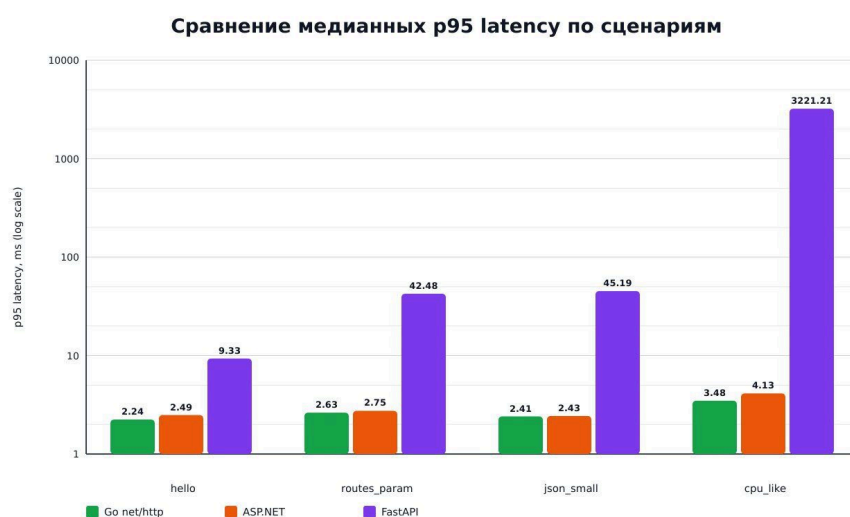


Рис. 4. Медианные p95 latency по сценариям (логарифмическая шкала)

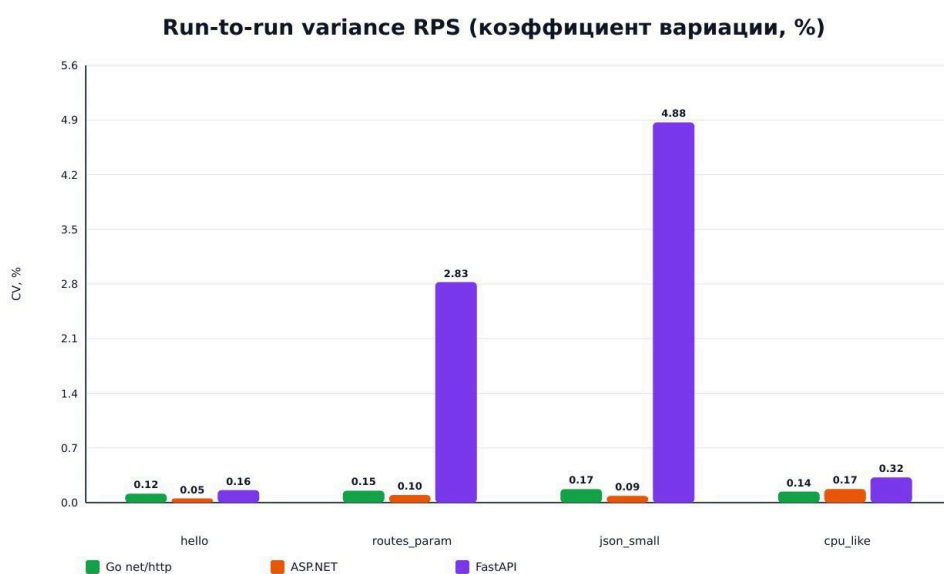


Рис. 5. Run-to-run variance RPS (коэффициент вариации, %)

4. Обсуждение результатов. Близость показателей Go и ASP.NET в обоих режимах нагрузки предсказуема для программ с компилируемым кодом и развитыми механизмами конкурентности. Оба хорошо справляются как с I/O-ожиданием, так и с вычислительными задачами.

Отставание FastAPI в `cpu_like` не означает принципиального изъяна фреймворка. В I/O-сценариях FastAPI показывает конкурентоспособные результаты. Проблема возникает именно тогда, когда работу определяют вычисления на уровне Python-кода: интерпретатор не успевает обслуживать очередь, и задержки растут лавинообразно [2]. При необходимости ситуацию можно частично улучшить, запустив несколько воркеров `uvicorn` или вынеся тяжёлые вычисления в отдельный нативный компонент.

Среди ограничений исследования следует отметить, что эксперимент проводился на одном хосте при фиксированной конфигурации окружения. Нагрузка без базы данных отражает поведение исключительно HTTP- и runtime-слоя. Влияние горизонтального масштабирования не изучалось.

5. Заключение. Проведённый эксперимент подтвердил, что корректное сравнение моделей параллелизма требует строгой унификации условий. Go и ASP.NET Core показали близкие значения пропускной способности и задержек в обоих режимах нагрузки. FastAPI обеспечил нулевую долю ошибок и конкурентоспособные результаты в I/O-сценариях, однако в CPU-сценарии существенно отстал из-за ограничений модели исполнения Python. Для CPU-нагруженных сервисов предпочтительны Go или ASP.NET Core; FastAPI рационален там, где приоритет отдан скорости разработки, а нагрузка носит преимущественно I/O-характер. Подготовленный шаблон эксперимента пригоден для повторного применения при обосновании выбора серверного стека.

Литература

1. The Go Programming Language Specification. URL: <https://go.dev/ref/spec> (дата обращения: 05.02.2026).
2. URL: <https://fastapi.tiangolo.com/> (дата обращения: 02.02.2026).
3. Python `asyncio` Documentation. URL: <https://docs.python.org/3/library/asyncio.html>. (дата обращения: 02.02.2026)
4. NET Core Documentation. URL: <https://learn.microsoft.com/aspnet/core/> (дата обращения: 03.02.2026).
5. k6 Documentation. URL: <https://k6.io/docs/>. (дата обращения: 06.02.2026)

D.S. Redchenkov, T.A. Samoilova
Smolensk State University

**COMPARISON OF PARALLELISM MODELS IN REST API:
GO, PYTHON/FASTAPI, ASP.NET UNDER LOAD**

Keywords: *parallelism, REST API, Go, FastAPI, ASP.NET Core, k6, throughput, latency, high-load services.*

Abstract. *This paper compares three models of concurrent HTTP request handling in REST API services: the goroutine-based Go net/http, event-driven Python/FastAPI, and thread-pool ASP.NET Core Minimal API. All three services share a common API contract, identical in-memory logic, and the same seed dataset. Load testing was performed with k6 in I/O and CPU scenarios across three repeated runs. Go and ASP.NET show near-parity in both workload types, while FastAPI falls significantly behind under CPU-intensive load due to Python runtime constraints. The paper provides practical stack selection recommendations and a reproducible benchmarking template.*

Т.А. Самойлова

Смоленский государственный университет

УДК 004.032.26

МЕТОД СОЗДАНИЯ ВЕБ-СЕРВИСА ДЛЯ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Ключевые слова: *веб-сервис, классификация изображений, реальное время, Python, CNN, FastAPI, React Native, Expo.*

В статье приведено описание метода выбора архитектуры, создания, обучения и сохранения модели сверточной нейронной сети (CNN) для многоклассовой классификации изображений. В том числе её загрузка в быстросействующее серверное веб-приложение, реализованное средствами фреймворка FastAPI, и разработка мобильного клиентского приложения средствами языка React Native для отправки изображения на сервер с целью определения его класса. Предложенный метод обеспечивает быстрое развертывание модели и взаимодействие в реальном времени подобных моделей с пользователями мобильных устройств. Проведен анализ экспериментальных результатов, подтверждающий высокую точность и производительность метода. Предложены возможные предметные области его использования.

1. Введение. В статье обоснован выбор средств автоматизации программирования для решения задач классификации изображений в реальном времени. Известно, что сверточные нейронные сети (CNN) [1]

эффективны для обработки изображений, особенно в плане точности [2]. Однако создание системы реального времени с мобильным клиентом и веб-сервисом для классификации изображений – сложная задача [3]. Она требует баланса между вычислительными требованиями и ограничениями ресурсов мобильных устройств. Современные средства автоматизации программирования обеспечивают решение проблемы разработки таких систем. В настоящей работе предложен метод создания трехуровневой архитектуры, в которой мобильное приложение (фронтенд), разработанное с помощью кроссплатформенного фреймворка React Native [4], обращается с запросами к веб-сервису (бэкенд), созданному с помощью Python-фреймворка FastAPI [5], взаимодействующего с предварительно обученной моделью CNN. Выбор CNN обусловлен тем, что она наиболее эффективна для обработки краев и текстур изображений, разбивая последние на мелкие части и обучаясь на этих деталях. Таким образом сеть может идентифицировать сложные признаки, что делает её эффективной для задач классификации медицинских изображений, дорожных знаков, элементов одежды, животных. Преимущества FastAPI заключаются в его высокой производительности и простоте интеграции с моделями глубокого обучения. Фреймворк поддерживает асинхронную работу, позволяя испытывать логику через OpenAPI (Swagger). В качестве мобильного клиента системы выбрана упрощающая разработку платформа Expo, построенная на основе React Native. Результаты экспериментов, полученные путем интеграции CNN, FastAPI и React Native, обсуждаются с использованием квалитетических оценок качества: точности, потерь и производительности. Целевые данные для обучения модели – набор Fashion-MNIST из лицензированной библиотеки tensorflow-datasets [6].

2. Выбор архитектуры модели CNN. Для этого проведен сравнительный анализ более пяти различных архитектур CNN. Главные критерии оценки качества – точность и потери. На рисунке 1 изображена архитектура выбранной модели, созданная на основе библиотеки TensorFlow [7] с использованием пакета Keras.

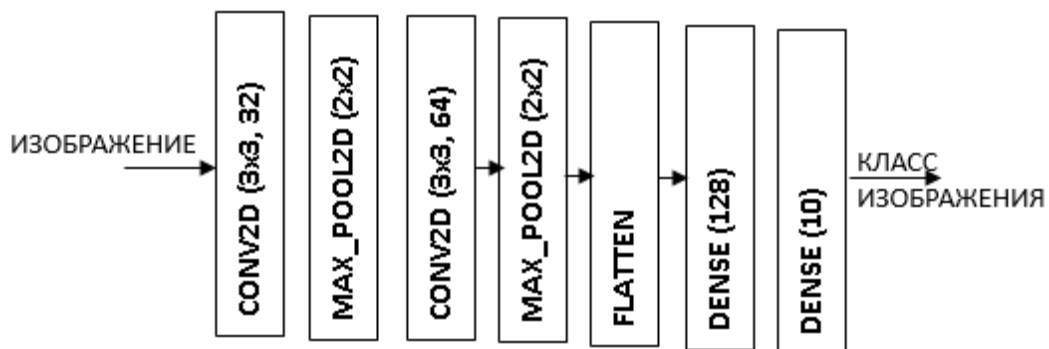


Рис. 1. Архитектура модели CNN для классификации изображений

Приведенная архитектура включает следующие слои:

- Conv2D – сверточный слой с 32 фильтрами, каждый размером (3x3).
- MaxPool2D – слой максимальной подвыборки (пулинга), снижающий размерность изображения.
- Conv2D – сверточный слой с 64 фильтрами, каждый размером (3x3).
- MaxPool2D – слой максимальной подвыборки.
- Flatten – сглаживающий слой, преобразует двумерные выходные данные сверточных слоев в одномерный вектор, необходимый для полносвязных (плотных) слоев.
- Dense – полносвязанный слой со 128 элементами, комбинирует признаки, выделенные предыдущими слоями, чтобы принять финальное решение о классе.
- Dense – полносвязанный слой для окончательной классификации по 10 категориям.

После создания модель была скомпилирована и обучена.

3. Развертывание модели на веб-сервисе. Для функционирования обученной модели CNN в реальном времени выполнено ее сохранение в файле, развертывание на веб-сервере FastAPI и предоставление к ней доступа мобильных клиентов через конечные точки. API-конечная точка доступа в FastAPI позволяет клиенту взаимодействовать с веб-сервисом по протоколу HTTP. Трехуровневая схема развертывания представлена на рисунке 2.

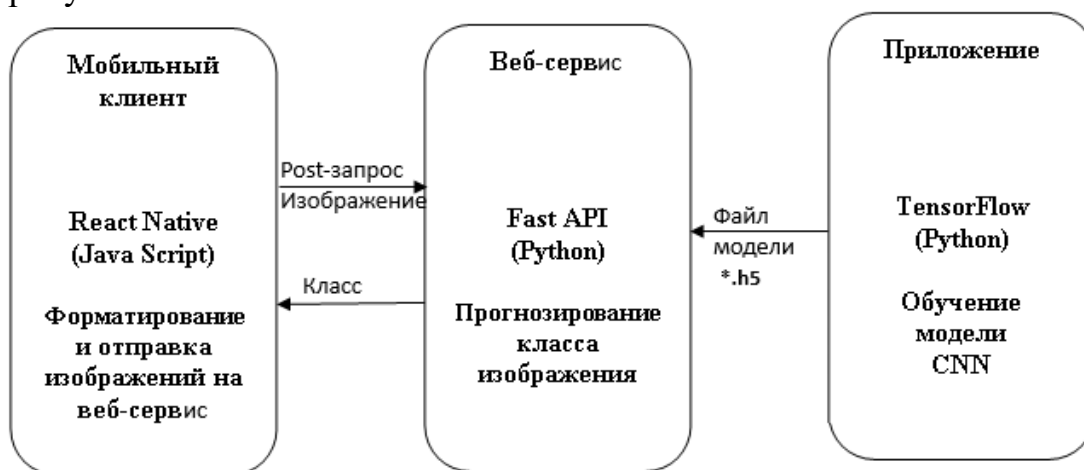


Рис. 2. Схема развертывания модели на веб-сервисе

На схеме видно, что обученная модель сохраняется в файле *.h5 и загружается в память веб-сервиса при его запуске. Для ее сериализации и сохранения используются Python-пакеты `scikit-learn` и `pickle`. Процесс включает выполнение функций `pickle.dump()` для сохранения модели в файл и `pickle.load()` для её извлечения. Веб-сервис загружает этот файл для прогнозирования класса. Прогнозирование выполняется средствами FastAPI путем создания конечной точки `@app.post`, которая принимает от мобильных клиентов изображения, преобразует их в необходимый

формат, запускает функцию прогнозирования и возвращает результат – классы изображений. Асинхронные возможности фреймворков FastAPI и React Native обеспечивают возможность реализации параллелизма и многопоточности как на рабочих станциях, включая мобильных клиентов, так и на серверах. Это становится возможным благодаря тому, что FastAPI запускает несколько рабочих процессов на всех доступных ресурсах (ядрах). Параллелизм в рамках одного процесса достигается за счет асинхронной передачи данных (async) с последующим ожиданием (await). Для предотвращения ошибок безопасности веб-сервис включает протокол CORS (Cross-Origin Resource Sharing), контролирующий взаимодействие клиентских приложений с сервером.

POST-запросы мобильного приложения React Native к конечным точкам FastAPI реализованы библиотекой Axios на платформе Expo, ускоряющей создание приложения. Исходное изображение предварительно кодируется и отправляется на сервер в структуре данных формы. Главные достоинства React Native – кроссплатформенность (один код для iOS и Android), отзывчивость, означающая адаптацию интерфейса под разные размеры экранов и обеспечение быстрой реакции на действия пользователей. В задаче классификации загруженное клиентом новое изображение предварительно обрабатывается так же, как и в процессе обучения.

4. Данные эксперимента. Эксперименты с моделью проведены на целевом наборе «Модная одежда», состоящем из 70 000 изображений размером 28x28 пикселей, относящихся к 10 категориям модной одежды. Метки соответствующих классов: 0 – футболка, 1 – брюки, 2 – пуловер, 3 – платье, 4 – пальто, 5 – сандалии, 6 – рубашка, 7 – кроссовки, 8 – сумка, 9 – ботильоны. На рисунке 3 представлен фрагмент изображений из набора:



Рис. 3. Фрагмент изображений целевого набора «Модная одежда»

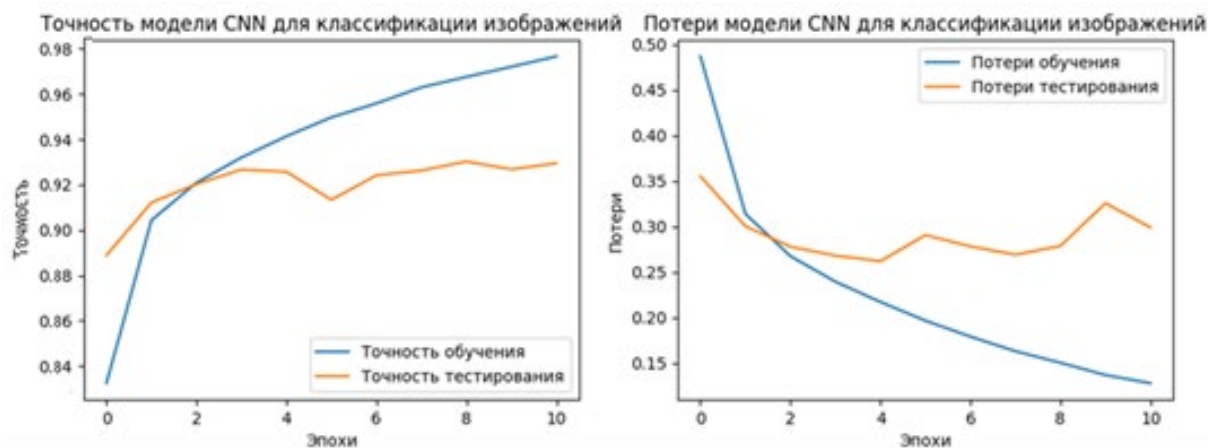
Данные разделяются на обучающий набор из 60 000 изображений и тестовый, включающий 10 000 изображений. После обучения выполняется оценка, насколько хорошо модель работает на данных, которые она не использовала для обучения – тестовом наборе.

5. Результаты эксперимента. Разработка системы классификации изображений в реальном времени включает четыре основных этапа, назначение которых – обучение и сохранение модели CNN, разворачивание её на веб-сервисе, тестирование производительности сервиса и выполнение POST-запроса к сервису с мобильного устройства для получения класса исходного изображения. Эксперимент проводился на рабочей станции со следующими характеристиками:

- центральный процессор 13th Gen Core i9 (24 ядра, из которых 8 высокой производительности, 32 потока);
- оперативная память 64 ГБ;
- SSD 2ТБ;
- графический процессор NVIDIA GeForce RTX 5060 Ti, выбран с учетом того, что используемые фреймворки часть операций выполняют на графических ускорителях.

Далее приведены результаты этапов эксперимента.

Первый этап – обучение и сохранение модели CNN. На рисунке 4 показаны графики точности и потерь в ходе обучения и тестирования модели на целевом наборе в среде PyCharm.



a)

b)

Рис. 4. Точность (a) и потери (b) модели CNN при обучении и тестировании

Приведенные графики демонстрируют высокую точность и небольшие потери модели. На обучающем наборе установившиеся значения точности на десятой эпохе составили 0.98, потерь – 0.1. На тестовом наборе значения точности – 0.92, потерь – 0.3. Время обучения модели на производительном процессоре и достаточно большом наборе данных – 3.8 минуты. Чтобы увидеть основную разницу в предсказаниях модели для разных классов, построена матрица ошибок (рисунок 5).

Диагональ матрицы представляют количества правильно предсказанных экземпляров тестовой выборки.

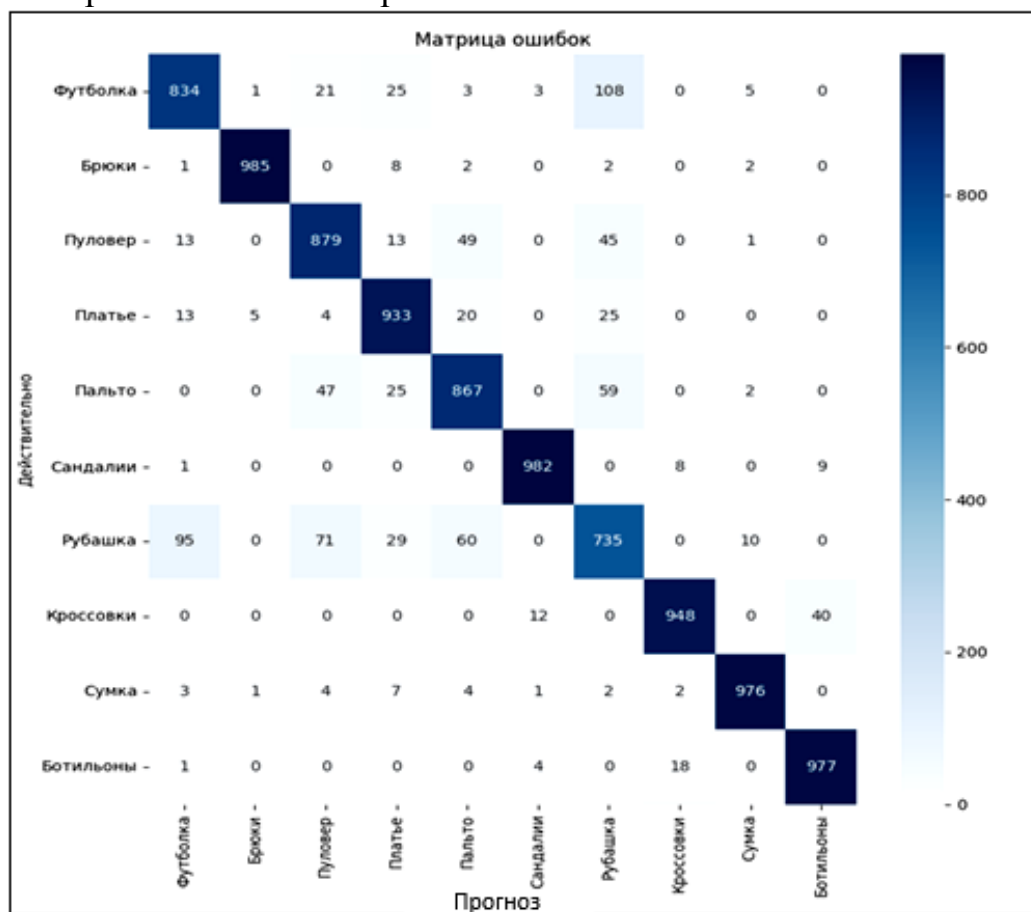


Рис. 5. Матрица ошибок модели CNN

Эксперимент показал, что результаты обучения удовлетворительные и модель сохранена.

Второй этап – разворачивание модели на веб-сервисе и его испытания. Для этого выполнена настройка конечной точки POST-запроса, которая принимает от мобильного устройства файл изображения, обрабатывает его в соответствии с требованиями модели к входным данным, осуществляет прогнозирование и возвращает результат – класс изображения и достоверность прогноза, количественно определяющую уверенность в прогнозе. Для испытаний работы конечной точки использован набор инструментов Swagger, обеспечивающий загрузку нового изображения и его классификацию непосредственно в браузере. Фрагмент окна браузера показан на рисунке 6.

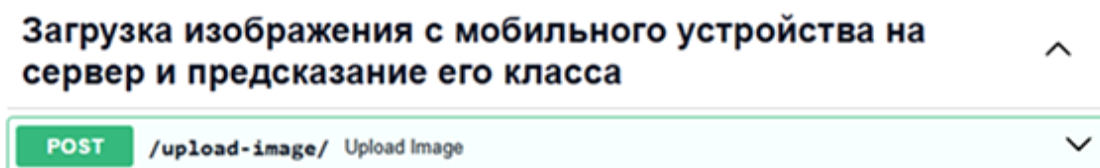


Рис. 6. Фрагмент Swagger для выполнения POST-запроса

Третий этап – тестирование производительности веб-сервиса. Выполнен фреймворком Locust, определяющим, как система будет функционировать под нагрузкой множества пользователей. Имитируя одновременную работу нескольких пользователей, locust.io предоставляет информацию о производительности веб-сервиса в виде числа запросов, которые он может обрабатывать в секунду для заданного числа клиентов (RPS – Requests Per Second). Чем выше значение RPS, тем лучше производительность системы. В таблице 1 предоставлены результаты оценки производительности для разного числа клиентов сервиса.

Таблица 1

Клиентов:	10	50	100	150	200	250	300	500	1000
RPS, ms	5,8	26	49	72	97	115	143	165	398

Из таблицы отчетливо видно, что веб-сервис может обрабатывать большой трафик, обеспечивая при этом приемлемое время отклика.

Четвертый этап – непосредственное взаимодействие веб-сервиса с удаленным мобильным клиентом. Приложение клиента разработано с использованием фреймворка Expo React Native, языка Java Script и среды Visul Studio Code. При этом Expo позволяет осуществлять запуск проекта в Android Studio, Xcode (iOS), на мобильном устройстве или браузере. Интерфейс приложения после обращения к веб-сервису изображен на рисунке 7.



Рис. 7. Интерфейс мобильного приложения для предсказания класса

5. Заключение. Интеграция CNN, быстросействующего веб-сервиса и мобильного приложения становится одним из основных способов решения проблем, связанных с созданием систем искусственного интеллекта, работающих в режиме реального времени. Чтобы применить CNN к практическим задачам, их необходимо обучить на конкретных наборах данных. Однако, поскольку масштаб этих данных, а с ним и время обучения, постоянно увеличивается, стало

необходимым использовать быстродействующий веб-сервис с предварительно сохраненной моделью. Поэтому предложенный в настоящей работе метод создания такого веб-сервиса современен и актуален. Проведен эксперимент, включающий четыре этапа разработки сервиса: обучение и сохранение модели CNN, разворачивание ее на веб-сервисе, тестирование производительности сервиса и его взаимодействие с мобильным приложением. Результаты демонстрируют высокую производительность системы при точности модели на тестовых данных свыше 92% и достоверности прогнозов до 99%. Это позволит осуществлять в реальном времени взаимодействие подобных моделей с пользователями мобильных устройств в таких областях, как розничная торговля, электронная коммерция, медицинская диагностика, транспорт, сельскохозяйственные технологии, социальные сети и ряд других.

Литература

1. Alzubaidi L. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions // Journal of big Data. 2021. Т. 8, №. 1. С. 53.
2. Elngar A. A. et al. Image classification based on CNN: a survey // Journal of Cybersecurity and Information Management. 2021. Т. 6, №. 1. С. 18-50.
3. Irtaza G. et al. Real-time Satellite Image Classification Using Convolutional Neural Networks for Earth Observation and Video Feed Analysis // Kashf Journal of Multidisciplinary Research. 2025. Т. 2, № 03. С. 204-216.
4. React Native // URL: <https://reactnative.dev/>
5. FastAPI // URL: <https://fastapi.tiangolo.com/>
6. Fashion-MNIST // URL: <https://github.com/zalandoresearch/fashion-mnist/>
7. TensorFlow // URL: <http://tensorflow.org>

T.A. Samoilova

Smolensk State University

ANALYSIS OF HYPERPARAMETER OPTIMIZATION METHODS IN MACHINE LEARNING SYSTEMS

Keywords: *web service, image classification, real-time, Python, CNN, FastAPI, React Native, Expo.*

Abstract. *This article describes a method for selecting the architecture, creating, training, and storing a convolutional neural network (CNN) model for multi-class image classification. This includes loading the model into a high-performance server-side web application implemented using the FastAPI framework and developing a mobile client application using React Native to send the image to the server for class determination. The proposed method enables rapid model deployment and real-time interaction between similar models and mobile device users. Experimental results are analyzed, confirming the accuracy and performance of the method. Potential application areas for its use are suggested.*

МОДЕЛЬ МУЛЬТИЭШЕЛОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗАПАСАМИ И ПОПОЛНЕНИЯ РЕСУРСОВ

Ключевые слова: *мультиэшелонные запасы, координация цепочки поставок, пополнение ресурсов, модель оптимизации, минимизация затрат.*

В статье исследуется координация в мультиэшелонных цепях поставок в условиях неопределенности спроса. Автор предлагает математическую модель оптимизации, учитывающую затраты на закупку, хранение, транспортировку и штрафы за дефицит. Для динамического пополнения ресурсов применяется марковский процесс принятия решений (MDP), позволяющий системе адаптироваться к изменениям среды. Модель интегрирует бюджетные и мощностные ограничения для минимизации общих издержек. Предложенный подход оптимизирует уровень запасов, обеспечивая баланс между экономической эффективностью и надежностью поставок.

1. Постановка задачи. В сфере производства товаров управление запасами является ключевым компонентом управления цепочками поставок, особенно в мультиэшелонных системах управления запасами. В условиях глобализации и усложнения цепочек поставок управление запасами сталкивается с повышенными трудностями из-за необходимости эффективной координации между звеньями цепочки на разных уровнях, такими как производители, оптовики, дистрибьюторы и розничные продавцы [1].

Глобализация приводит к удлинению цепочек поставок с участием различных поставщиков и рынков, что повышает вероятность сбоев и нестабильности спроса. Кроме того, сложность многоэшелонных цепочек поставок часто приводит к трудностям в прогнозировании структуры спроса и управлении запасами на взаимосвязанных узлах.

Эти факторы требуют применения передовых стратегий для обеспечения экономической эффективности, сокращения отходов и поддержания уровня обслуживания в условиях динамической неопределенности в цепочке поставок.

Многоэшелонное управление запасами подразумевает координацию и оптимизацию запасов в многоуровневых сетях поставок. Его цель – координировать стратегии управления запасами на нескольких узлах, чтобы оптимизировать общий уровень запасов и затраты в цепочке поставок. Цепочка поставок товаров повседневного спроса охватывает

весь процесс – от поставки сырья, производства и дистрибуции до конечного потребителя [2]. Многоуровневое управление запасами особенно важно в цепочке поставок наиболее востребованных ресурсов, поскольку позволяет эффективно координировать запасы на разных уровнях, сокращать избыточные и дефицитные запасы, а также повышать гибкость и оперативность цепочки поставок. Цепочки поставок таких ресурсов также значительно выигрывают от цифровой трансформации, которая улучшает прогнозирование спроса и координацию между уровнями [3].

Если говорить подробнее, процесс пополнения запасов в многоэшелонной цепочке поставок выглядит следующим образом.

Сеть поставок состоит из нескольких уровней и обычно включает в себя производителей, оптовиков, дистрибьюторов и розничных продавцов. Пополнение запасов на каждом уровне зависит от динамических прогнозов спроса и уровня запасов на последующих этапах. Процесс пополнения запасов можно разделить на три основных этапа. Перечислим их.

1. Прогнозирование спроса: уровень запасов в розничных магазинах отслеживается в режиме реального времени, а прогнозы спроса периодически обновляются на основе исторических данных о продажах и предиктивной аналитики.

2. Решение о пополнении запасов. На основе прогнозов спроса и уровня запасов вышестоящие звенья (например, дистрибьюторы или оптовики) принимают решение о количестве пополняемых запасов, чтобы обеспечить достаточный запас на нижестоящих звеньях, минимизировав при этом затраты на хранение и транспортировку.

3. Транспортировка и распределение. Запасы доставляются на нижестоящие узлы по оптимизированным маршрутам и графикам, чтобы минимизировать задержки и затраты.

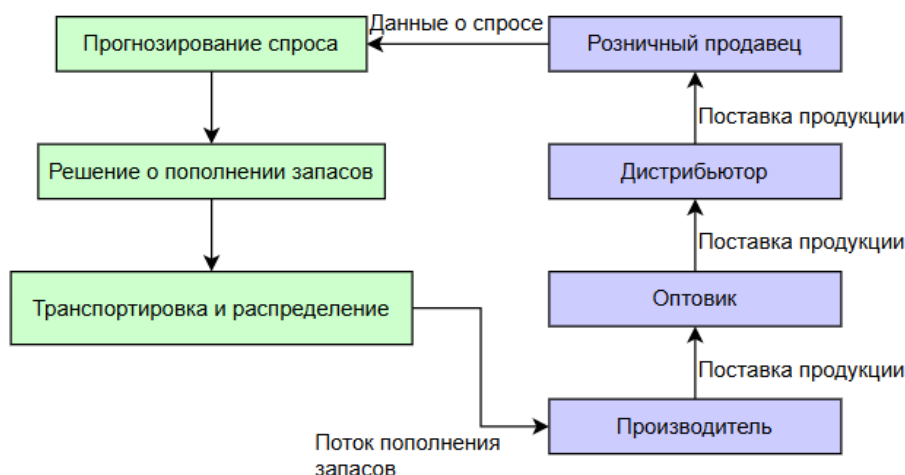


Рис. 1. Схема процесса пополнения запасов в типичной многоэшелонной сети поставок ресурсов

На рисунке 1 показано визуальное представление процесса пополнения запасов в типичной многоэшелонной сети поставок ресурсов.

Цепочка поставок ресурсов, включающая производителей, оптовиков, дистрибьюторов, розничных продавцов и покупателей, зависит от слаженной работы и координации на всех уровнях, что обеспечивает стабильность поставок [4]. Однако из-за задержек с передачей информации между звеньями, неточного прогнозирования спроса и неэффективного управления запасами дефицит ресурсов по-прежнему является распространенным явлением. Менеджеры по управлению цепочками поставок должны быстро выявлять и устранять причины дефицита, обеспечивая своевременные поставки альтернативных товаров для удовлетворения потребностей клиентов. Дисбаланс запасов на разных уровнях может привести к более масштабным последствиям для всей цепочки поставок, что повлечет за собой увеличение общих затрат на систему и снижение эффективности производственных цепочек.

Причины перебоев в поставках продукции многообразны: это и резкий рост спроса, и сбои в координации на различных этапах цепочки поставок, и непредсказуемые внешние факторы, такие как стихийные бедствия. Дефицит, вызванный этими факторами, не только увеличивает операционные расходы, но и может серьезно повлиять на эффективность всей цепочки поставок. Поэтому эффективное многоэтапное управление запасами имеет большое значение для сокращения дефицита и повышения гибкости цепочки поставок. Предприятия должны уметь оптимизировать уровень запасов на разных этапах цепочки поставок, обеспечивая достаточный резерв для удовлетворения внезапных изменений спроса и при этом максимально сокращая расходы [5].

Далее рассмотрим сложную мультиэшелонную систему управления запасами и пополнением ресурсов (СУЗПР). Эта система включает в себя множество звеньев (например, производителей, оптовиков, дистрибьюторов и розничных продавцов), которым необходимо координировать уровень запасов и решение о пополнении ассортимента для общей оптимизации цепочки поставок. Система работает в режиме временных интервалов $1, 2, \dots, T$, где каждый из которых представляет собой определенный период (например, одну неделю или один месяц).

2. Метод решения задачи. Для решения проблем с координацией между звеньями цепочки поставок моделируется многоуровневая задача оптимизации запасов. Кроме того, для обеспечения баланса между оптимизацией и риском дефицита запасов вводятся штрафные функции и весовые коэффициенты.

Стоимость закупки напрямую зависит от количества заказанных ресурсов. Она выражается как

$$\sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} \alpha_{f,l} \gamma_{f,l}^{(t)} x_{f,l}(t), \quad (1)$$

где $\alpha_{f,l} \in [0,1]$ – весовой параметр для ресурсов f на уровне l , и $r_{f,l}(t)$ представляет собой удельную стоимость закупки ресурсов f на уровне l за период t .

Стоимость хранения связана с избыточными запасами и выражается как

$$\sum_{t=1}^T \sum_{l=1}^L M_{f,l}(v_{f,l}(t), m_{f,l}(t)), \quad (2)$$

где $v_{f,l}(t)$ – это избыточные запасы ресурсов f на уровне l в момент времени t , а $m_{f,l}(t)$ – это стоимость хранения ресурсов f на уровне l в течение времени t .

Запасы ресурсов f на уровне l за время t рассчитываются как

$$v_{f,l}(t) = v_{f,l}(t-1) - d_{f,l}(t) - e_{f,l}(t) + x_{f,l}(t), \quad (3)$$

где $v_{f,l}(t-1)$ – это избыточные запасы за предыдущий период, $d_{f,l}$ – величина спроса, и $e_{f,l}(t)$ – это количество ресурса с истекающим сроком годности.

Стоимость транспортировки зависит от объема пополнения запасов и расстояния транспортировки и выражается как

$$\sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} \gamma_{f,l} x_{f,l}(t) \cdot d_{f,l}, \quad (4)$$

где $\gamma_{f,l}$ – коэффициент транспортных расходов на единицу расстояния, $x_{f,l}(t)$ – расстояние транспортировки товаров f на уровне l .

Функция расчета стоимости дефицита направлена на предотвращение дефицита запасов, который выражается как

$$\sum_{t=1}^T \sum_{l=1}^L \mathcal{G}(v_{f,l}(t)), \quad (5)$$

где $\mathcal{G}(v_{f,l}(t))$ зависит от разницы между уровнем запасов и минимальным требованием к запасам.

Целевая функция: цель состоит в том, чтобы минимизировать общую стоимость, выраженную как

$$\min \sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} \left[\alpha_{f,l} \gamma_{f,l}(t) x_{f,l}(t) + M_{f,l}(v_{f,l}(t), m_{f,l}(t)) + \theta(v_{f,l}(t)) + \gamma_{f,l} x_{f,l}(t) d_{f,l} \right]. \quad (6)$$

Для обеспечения реализуемости модели накладываются следующие ограничения:

Ограничения по вместимости состоят в том, что объем пополнения не превышает доступную вместимость хранилища.

$$C_{f,l}(t) \geq v_{f,l}(t) = v_{f,l}(t-1) - d_{f,l}(t) - e_{f,l}(t) + x_{f,l}(t), \quad \forall f \in F, t=1,2,\dots,T. \quad (7)$$

Бюджетные ограничения предполагают, чтобы расходы на приобретение ресурсов и пополнение запасов не выходили за рамки общего бюджета.

$$\sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} x_{f,l}(t) \gamma_{f,l}(t) \leq B, \quad (8)$$

где B – общий бюджет, а

$$x_{f,l}(t) \gamma_{f,l}(t) \leq B_{f,l}, \quad \forall f \in F; \quad (9)$$

Ограничения по грузоподъемности состоят в том, что объем перевозимого груза не превышает максимальную грузоподъемность.

$$\sum_{l=1}^L \sum_{f \in F} x_{f,l}(t) \leq Y_{\max}(t), \quad \forall t=1,2,\dots,T, \quad (10)$$

где $Y_{\max}(t)$ обозначает максимальную грузоподъемность;

Ограничения по пополнению запасов предполагают, что количество пополняемых запасов соответствует заданным ограничениям:

$$p_{f,l}(t) \leq x_{f,l}(t) \leq \bar{p}_{f,l}(t), \quad \forall f \in F, t=1,2,\dots,T, \quad (11)$$

где $p_{f,l}(t), \bar{p}_{f,l}(t)$ – это нижний и верхний пределы пополнения соответственно.

3. Динамическая многоуровневая модель оптимизации пополнения запасов. В этой модели ресурсов может быть сформулирована следующим образом:

$$\min_x \sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} \left[\alpha_{f,l} \gamma_{f,l}(t) x_{f,l}(t) + M_{f,l}(v_{f,l}(t), m_{f,l}(t)) + \theta(v_{f,l}(t)) + \gamma_{f,l} x_{f,l}(t) d_{f,l} \right]$$

при следующих ограничениях,

$$C_{f,l}(t) \geq v_{f,l}(t) = v_{f,l}(t-1) - d_{f,l}(t) - e_{f,l}(t) + x_{f,l}(t) \leq C_{f,l}(t), \quad \forall f \in F, t=1,2,\dots,T,$$

$$\sum_{t=1}^T \sum_{l=1}^L \sum_{f \in F} x_{f,l}(t) \gamma_{f,l}(t) \leq B,$$

$$x_{f,l}(t)\gamma_{f,l}(t) \leq B_{f,l}, \quad \forall f \in F,$$

$$\sum_{l=1}^L \sum_{f \in F} x_{f,l}(t) \leq Y_{\max}(t), \quad \forall t=1,2,\dots,T,$$

$$p_{f,l}(t) \leq x_{f,l}(t) \leq \bar{p}_{f,l}(t), \quad \forall f \in F, t=1,2,\dots,T.$$

В многоуровневой системе управления запасами необходимо в каждый момент времени t принимать решение о пополнении запасов ресурсов, несмотря на неопределенность в отношении будущего спроса. Эта неопределенность может привести к дефициту на следующей неделе, если предприятия будут тратить значительные средства на пополнение запасов, не используя их полностью из-за истечения срока годности. И наоборот, если уровень запасов слишком низкий, это может привести к дефициту.

Чтобы решить эти проблемы, целесообразно использовать метод на основе машинного обучения для преодоления следующих трудностей.

1. Влияние неопределенности спроса на принятие решений в каждом временном интервале.

2. Способность модели адаптироваться к меняющимся условиям и принимать самостоятельные решения.

В задаче многоуровневой оптимизации запасов марковский процесс принятия решений (Markov Decision Process, MDP) обеспечивает основу для систематизации процесса принятия решений. Модель разбивает сложную многоуровневую задачу управления цепочками поставок на ряд последовательных процессов принятия решений, каждый из которых соответствует определенному временному периоду и уровню цепочки поставок.

Состояние системы s_t описывает ключевые переменные многоуровневой системы управления запасами на момент времени t . Состояние системы включает в себя следующие аспекты:

1. Уровень запасов $v_{f,l}(t)$: текущий уровень запасов товаров повседневного спроса f на уровне l в момент времени t .

2. Спрос $d_{f,l}$: прогнозируемый спрос на товары повседневного спроса f на уровне l и в момент времени t .

3. Параметры стоимости:

- стоимость пополнения $\gamma_{f,l}(t)$ запасов f на уровне l за время t ;
- стоимость хранения $m_{f,l}(t)$ запасов f на уровне l в момент времени t .

4. Ограничения пропускной способности:

- емкость для хранения $C_{f,l}(t)$ – максимальная вместимость склада для запасов f на уровне l и в момент времени t ;

- минимальные требования к запасам $C_{f,l}^*(t)$ – минимальный необходимый запас товаров повседневного спроса f на уровне l для предотвращения дефицита.

5. Бюджет и транспортные возможности:

- бюджет, выделенный на приобретение запасов f на уровне l ;
- транспортная пропускная способность $\Upsilon_{\max}(t)$ – максимальная пропускная способность на момент t .

Таким образом, состояние системы s_t может быть представлено в виде многомерного вектора, содержащего всю необходимую информацию о запасах и ресурсах, а также уровнях цепочки поставок:

$$s_t = \left\{ v_{f,l}(t), d_{f,l}(t), \gamma_{f,l}(t), m_{f,l}(t), C_{f,l}(t), C_{f,l}^*(t), B_{f,l}, \Upsilon_{\max}(t) \right\}, \forall f \in F, l \in L.$$

На каждом временном шаге t система должна определить количество пополнения для каждого уровня l и товара повседневного спроса f . Набор действий A_t определяется как

$$A_t = \left\{ a_{f,l}(t) \mid a_{f,l}(t) \in R^+, \forall f \in F, l \in L \right\},$$

где $a_{f,l}(t)$ представляет собой объем пополнения запасов f на уровне l и в момент времени t , зависящий от текущего состояния s_t и ограничений, таких как бюджет и транспортные возможности.

Функция вознаграждения R_t отражает экономическую выгоду системы от выполнения определенного действия с учетом затрат и дефицита ресурсов. Функция вознаграждения определяется как

$$R_t = - \sum_{f \in F} \sum_{l \in L} \left[\alpha_{f,l}(t) \gamma_{f,l}(t) x_{f,l}(t) + M_{f,l}(v_{f,l}(t)), m_{f,l}(t) + \mathcal{G}(v_{f,l}(t)) + \gamma_{f,l} x_{f,l}(t) d_{f,l} \right],$$

где $\mathcal{G}(v_{f,l}(t))$ – функция штрафа, которая наказывает за уровень запасов ниже минимально необходимого для поддержания стабильности системы.

В рамках MDP состояние системы s_t переходит в следующее состояние s_{t+1} после выполнения действия A_t .

Переход между состояниями определяется вероятностью $P(s_{t+1} | s_t, A_t)$, которая отражает вероятность достижения состояния s_{t+1} , учитывая текущее состояние s_t и действие A_t .

Основная цель MDP – определить оптимальную политику π^* , которая максимизирует совокупное вознаграждение за период T . Цель оптимизации выражается как

$$\max_{\pi} E \left[\sum_{t=1}^T R_t \right],$$

где в прогнозе учитываются стохастический характер спроса и другие факторы неопределенности в цепочке поставок.

Благодаря такой оптимизации система может принимать оптимальные решения о пополнении запасов в многоуровневой цепочке поставок, обеспечивая удовлетворение спроса, минимизируя общие затраты на пополнение запасов и снижая риск дефицита ресурсов.

Литература

1. Zhang Y., Chai Y., & Ma L. (2021). Research on Multi-Echelon Inventory Optimization for Fresh Products in Supply Chains. *Sustainability*, 13(11), 6309. <https://doi.org/10.3390/su13116309>.
2. Rachman R., Tingey J., Allmendinger R., Shukla P., & Pan W. (2025). Reinforcement Learning for Multi-Objective Multi-Echelon Supply Chain Optimisation. *arXiv*. <https://doi.org/10.48550/arxiv.2507.19788>.
3. Преференциальный алгоритм управления процессами ресурсного обеспечения и пополнения запасов / А.В. Высочкин [и др.] // *Естественные и технические науки*, 2019, №2. С.195-196.
4. Si Thu Thant Sin. «The Parallel Processing Approach to the Dynamic Programming Algorithm of Knapsack Problem». 2021. IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia, 2021. P. 2252-2256. doi: 10.1109/ElConRus51938.2021.9396489.
5. Kolpakov R.M., Posypkin M.A., Si Thu Thant Sin. Complexity of solving the Subset Sum problem with the branch-and-bound method with domination and cardinality filtering. // *Automation and Remote Control*. 2017. Vol. 78, No. 3. P. 463–474.

Si Thu Thant Sin

National Research University of Electronic Technology (MIET)

MODEL OF A MULTI-ECHELON INVENTORY MANAGEMENT AND REPLENISHMENT SYSTEM

Keywords: *multi-echelon inventory, supply chain coordination, resource replenishment, optimization modeling, cost minimization.*

Abstract. *This article examines coordination in multi-echelon supply chains under demand uncertainty. The author introduces a mathematical optimization model that incorporates procurement, storage, and transportation costs, as well as shortage penalties. To manage dynamic resource replenishment, a Markov Decision Process (MDP) is employed, allowing the system to adapt to changing conditions autonomously. The model integrates budget and capacity constraints to minimize total system costs. This approach optimizes inventory levels, ensuring a balance between economic efficiency and supply chain reliability.*

МОДЕЛИРОВАНИЕ ЦИФРОВЫХ ДВОЙНИКОВ ДЛЯ СИСТЕМ МУЛЬТИЭШЕЛОННОГО УПРАВЛЕНИЯ ПРОЦЕССАМИ РЕСУРСНОГО ОБЕСПЕЧЕНИЯ

Ключевые слова: мультиэшелонная цепочка поставок, цифровой двойник, управление ресурсами, нейтрософская логика, многоцелевая оптимизация.

В статье представлена многоцелевая модель оптимизации на основе цифровых двойников для умных мультиэшелонных цепочек поставок, специально предназначенная для управления ресурсами в условиях неопределенности. Благодаря интеграции нейтрософской логики, модель эффективно учитывает неопределенность спроса, затрат и мощностей, используя параметры истины, неопределенности и ложности. Фреймворк направлен на балансировку противоречивых целей: минимизацию затрат, максимизацию уровня обслуживания и снижение воздействия на окружающую среду. Для решения этой сложной NP-трудной задачи применяется гибридная метаэвристика GWGO (сочетание оптимизатора серого волка и генетического алгоритма), демонстрирующая превосходную сходимость и качество решений по сравнению с автономными алгоритмами.

Постановка задачи. Повышение эффективности процессов мультиэшелонного управления процессами ресурсного обеспечения часто включает несколько противоречивых целей, таких как минимизация затрат, максимизация уровня обслуживания и снижение воздействия на окружающую среду. Традиционные модели с одной целью недостаточны для реальных применений, где компромиссы между целями должны быть тщательно организованы. Методы многоцелевой оптимизации (МОО), направленные на поиск парето-оптимальных решений, стали стандартным подходом для сложных задач управления процессами ресурсного обеспечения. В литературе применялись различные методы МОО, включая взвешенные методы суммы, методы с ε -ограничениями и метаэвристические алгоритмы. Среди них метаэвристические алгоритмы приобрели значительную популярность благодаря своей способности решать крупномасштабные, нелинейные и невыпуклые задачи. Генетический алгоритм (GA), оптимизация роя частиц (PSO) и оптимизатор серого волка (GWO) – одни из широко используемых метаэвристических алгоритмов в исследованиях цепочки поставок. Например, Говиндан и др. применили

многоцелевую GA для оптимизации замкнутой цепочки поставок в процессах ресурсного обеспечения с учётом экономических и экологических целей [1]. Аналогично, Dubey и др. использовали PSO для выбора поставщиков и распределения заказов в условиях неопределённости [2]. В последнее время были предложены гибридные метаэвристики для повышения качества раствора и скорости сходимости. Например, гибридный подход, объединяющий GWO и GA, был представлен в работе [3] для решения крупномасштабных задач проектирования логистических сетей. Гибридный метод использовал исследовательские возможности GWO и возможности эксплуатации GA, что обеспечило лучшую производительность по сравнению с автономными алгоритмами. Несмотря на широкое использование метаэвристик в МОО, количество исследований по применению гибридных метаэвристик в цепочках поставок на основе цифровых двойников, особенно в условиях неопределённости, ограничено.

В последние годы технология цифровых двойников была предложена как один из ключевых подходов для повышения производительности, мониторинга и интеллектуального принятия решений в производственной и перерабатывающей промышленности. В частности, в работе [4] было показано, что применение цифрового двойника в управлении процессом плавки стали может преодолеть ограничения, вызванные суровыми условиями окружающей среды и слабыми возможностями мониторинга, а также повысить уровень виртуальной визуализации и управления процессами. В работе [5] была представлена инновационная модель коммуникации, основанная на виртуальных сенсорах, которая обеспечивает синхронизацию и обмен данными в многомерных цифровых двойниках. Была разработана комплексная структура для проектирования цифровых двойников в производственных системах, которая решает такие проблемы, как высокая стоимость, сложность проектирования и надёжность данных, используя искусственный интеллект в производстве, морфологический анализ и нечеткое принятие решений (Fuzzy TOPSIS).

Методология fuzzy TOPSIS представлена ниже.

1. Нормализация нечетких чисел.

Для каждой альтернативы x и критерия j нормированное нечеткое число μ_{ij} можно вычислить как

$$\mu_{ij} = \frac{\max(x_j) - \min(x_j)}{\max(x_j) - \min(x_j)}, \quad (1)$$

где x_{ij} – производительность альтернативы x по критерию j , $\min(x_j)$ и $\max(x_j)$ – это минимальные и максимальные значения производительности для критерия j соответственно.

2. Взвешенная нормализованная нечеткая матрица решений. Взвешенную нормированную нечеткую матрицу принятия решений R можно получить, умножив каждое нормированное нечеткое число μ_{ij} по соответствующему весу w_j , а затем выполнить суммирование по всем критериям:

$$R_{ij} = \mu_{ij} \times w_j. \quad (2)$$

3. Идеальные и антиидеальные решения. Идеальные и антиидеальные решения можно получить следующим образом:

$$A^* = \left(\max(R_{ij}) \right) \text{ for } i=1, \dots, n, \quad (3)$$

$$A^- = \left(\min(R_{ij}) \right) \text{ for } i=1, \dots, n, \quad (4)$$

где A^* – идеальное решение, A^- – антиидеальное решение, а n – количество альтернатив.

4. Расстояние между идеальными и антиидеальными решениями. Расстояние альтернативы x до идеальных и антиидеальных решений можно вычислить с помощью подходящей меры расстояния, такой как евклидово расстояние или манхэттенское расстояние:

$$Di^+(x) = \left[\sum_j (R_{ij} - A_j)^2 \right]^{0.5} \text{ for } j=1, \dots, m, \quad (5)$$

$$Di^-(x) = \left[\sum_j (R_{ij} - A_{j-})^2 \right]^{0.5} \text{ for } j=1, \dots, m, \quad (6)$$

где $Di^+(x)$ – расстояние альтернативы x до идеального решения; $Di^-(x)$ – расстояние альтернативы x до антиидеального решения, m – количество критериев, а A_j и A_{j-} – это j -тые значения критериев идеальных и антиидеальных решений соответственно.

5. Относительная близость к идеальным решениям. Относительная близость альтернативы x к идеальному решению может быть вычислена следующим образом:

$$C_i(x) = \frac{D_i^-(x)}{D_i^+(x) + D_i^-(x)}, \quad (7)$$

где $C_i(x)$ – относительная близость альтернативы x к идеальному решению.

В целом эти исследования показывают, что цифровые двойники благодаря их гибким возможностям признаны преобразующей технологией для повышения операционной эффективности, улучшения принятия решений и повышения интеллекта систем в различных производственных областях.

Исследуемая схема ресурсного обеспечения включает несколько взаимосвязанных уровней: поставщиков, производственных центров, распределительных центров, розничных торговцев и предприятий по переработке. Основные цели:

- минимизация общих затрат, включая производство, транспортировку и расходы на запасы;
- максимизация уровня сервиса, обеспечивая своевременную доставку и удовлетворённость клиентов.

Сеть работает при неопределённом спросе, транспортных затратах и мощностях. Эти неопределённости моделируются с помощью параметров, отражающих истинность (Т), неопределённость (I) и ложность (F) входных данных.

В этом исследовании рассматриваются следующие гипотезы:

- все требования клиентов должны быть удовлетворены без недостатка;
- расположение объектов цепочки поставок заранее определено;
- параметры, такие как спрос, затраты и мощности, выражаются нейтрософическими множествами.

Схематическое представление интеллектуальной цепочки поставок на основе цифровых двойников представлено на рисунке 1.

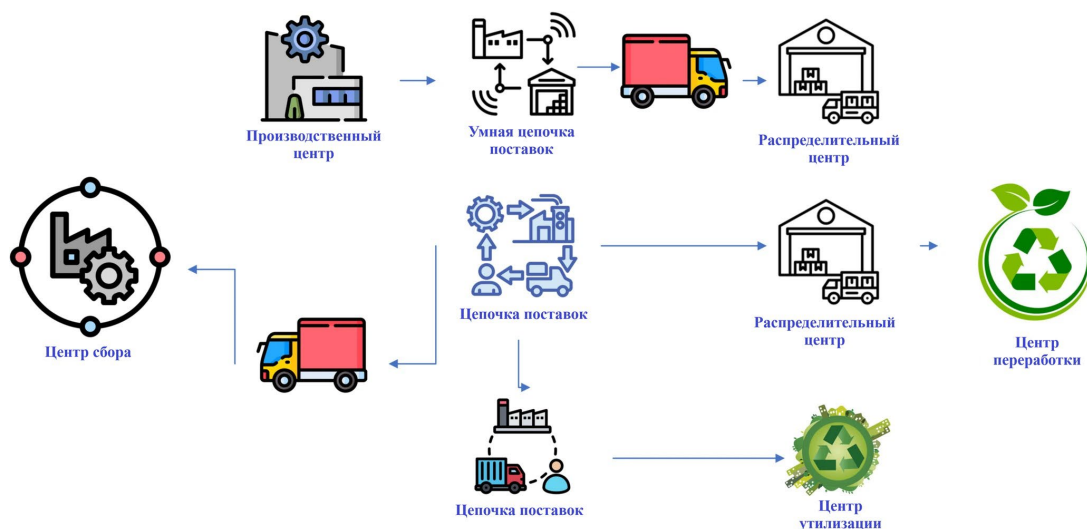


Рис. 1. Схематическое представление интеллектуальной цепочки поставок на основе цифровых двойников

Для облегчения практического внедрения предлагаемой нейрософической многоцелевой модели на основе цифровых двойников, особенно для малых и средних предприятий (МСП), мы описываем ключевые этапы, необходимые для её внедрения. Эти шаги разработаны так, чтобы быть простыми и управляемыми, внушать уверенность в целесообразности применения модели.

Настройка цифрового двойника:

1. Определение критически важных компонентов цепочки поставок, требующих мониторинга в реальном времени (например, поставщики, производственные центры, склады).

2. Развертывание IoT-датчиков для сбора данных в реальном времени (температура, уровни запасов, статус транспорта).

3. Разработка цифровой системы двойников, синхронизирующейся с программным обеспечением для планирования ресурсов предприятия (ERP).

Моделирование неопределённости с нейтрософической логикой:

4. Определение ключевых неопределённых параметров (например, спрос, затраты, сроки поставки) и классификация их с помощью нейтрософических множеств.

5. Проектирование эргономичного интерфейса, где менеджеры цепочек поставок могут корректировать значения истинности (Т), неопределённости (I) и ложности (F) на основе исторических данных.

Выполнение оптимизации:

6. Выбор подходящего алгоритма оптимизации (например, GA, PSO, GWO или предлагаемый гибридный GWGO).

7. Коррекция гиперпараметров в зависимости от размера цепочки поставок и вычислительной мощности.

8. Проведение анализа чувствительности для обеспечения устойчивости модели.

Ниже представлена разработанная нейтрософическая многоцелевая модель цифровых двойников для процессов ресурсного обеспечения.

$$MinTC = \sum_{t \in T} \left(\sum_{i \in I, j \in J} C_{ij} \cdot x_{ijt} + \sum_{j \in J, k \in K} C_{jk} \cdot y_{jkt} + \sum_{j \in J} H_j \cdot h_{jt} \right), \quad (8)$$

$$MaxSL = \frac{\sum_{k \in K, t \in T} \text{ДоставленныйЗапасЗаказов}}{\sum_{k \in K, t \in T} D_{kt}}, \quad (9)$$

$$MinEI = \sum_{t \in T} \left(\sum_{i \in I, j \in J} E_{ij} \cdot x_{ijt} + \sum_{j \in J, k \in K} E_{jk} \cdot x_{jkt} \right), \quad (10)$$

$$S \cdot t \sum_{j \in J} x_{ijt} \leq S_{it} \forall i \in I, t \in T, \quad (11)$$

$$\sum_{i \in I} x_{ijt} + h_{j(t-1)} = \sum_{k \in K} y_{ikt} + h_{jt}, \forall j \in J, t \in T, \quad (12)$$

$$\sum_{j \in J} y_{jkt} \geq D_{kt}, \forall k \in D, t \in T, \quad (14)$$

$$C_{ij} = C_{ij}^T \cdot (1 + \theta^I - \theta^F), \forall i \in I, j \in J \quad (15)$$

$$x_{ijt}, y_{jkt}, h_{jt} \geq 0 \forall i, j, k, t \quad (16)$$

$$h_{jt} \leq \text{СрокХранения}_j \forall j \in J, t \in T \quad (17)$$

где:

I – набор поставщиков;

J – набор объектов (например, склады, перерабатывающие центры);

K – группа продавцов / клиентов;

T – набор временных периодов;

D_{kt} – спрос со стороны продавца k в момент t ;

S_{it} – мощность поставщика i в момент t ;

C_{ij} – стоимость транспортировки товаров от поставщика i до объекта j ;

C_{jk} – стоимость транспортировки товаров от объекта j до продавца k ;

E_{ij} , E_{jk} – коэффициенты воздействия на окружающую среду для транспорта;

H_j – стоимость хранения на объекте j ;

θ – нейтрософический параметр неопределённости (истинность T , неопределённость I , ложность F);

x_{ijt} – объем ресурса, доставленный от поставщика i на объект j в момент t ;

y_{jkt} – объём, перевезённый с объекта j на склад k в момент t ;

h_{jt} – уровень инвентаря на объекте j в момент t .

Целевая функция (1) минимизирует общую стоимость. Целевая функция (2) максимизирует уровень обслуживания. Функция цели (3) минимизирует воздействие на окружающую среду. Уравнение (4) определяет мощность снабжения в цепочке поставок. Уравнение (5) отражает материальный баланс потока объекта. Ограничение (6) представляет собой удовлетворение спроса. Уравнение (7) представляет собой нейтрософические ограничения. Уравнения (8) определяют требования к неотрицательности переменных, а уравнение (9) определяет требования к безопасности пищевых продуктов и срокам хранения.

В реальных условиях цепочки поставок многие критически важные параметры – такие как спрос, транспортные затраты и производственные мощности – подвержены неопределённости. Традиционные вероятностные модели часто предполагают фиксированное распределение для этих неопределённостей, которые могут недостаточно отражать неопределённость и противоречивую информацию. Для решения этой проблемы мы включаем нейтрософическую логику, которая представляет неопределённость с помощью трёх независимых компонентов: истины (T), неопределённости (I) и лжности (F). Уравнение (7) включает эти параметры для динамической корректировки оценки стоимости и

повышения точности принятия решений в условиях неопределённости. Этот метод особенно полезен в цепочках поставок с высокой волатильностью и ограниченной доступностью данных, где классические вероятностные модели могут не дать надёжных инсайтов. При применении нейтрософического моделирования неопределённости предлагаемая структура гарантирует, что оптимизация цепочки поставок учитывает известную изменчивость и неопределённости, возникающие из-за непредсказуемой динамики рынка, что приводит к более устойчивым и адаптивным процессам принятия решений.

Представленная модель интегрирует технологии цифровых двойников, динамически обновляя D_{kt} , S_{it} , C_{ij} , C_{jk} и другие параметры на основе данных в реальном времени с физического уровня цепочки поставок. Нейтрософическая логика применяется для решения неопределённостей, присущих цепочке поставок.

При надёжной оптимизации цель состоит в том, чтобы оптимизировать наихудший сценарий для достижения целей в условиях неопределённости.

$$MinTC = \sum_{t \in T} \left(\max_{C_{ij} \in u_{C_{ij}}} \sum_{i \in I, j \in J} C_{ij} \cdot x_{ijt} + \max_{C_{jk} \in u_{C_{jk}}} \sum_{j \in J, k \in K} C_{jk} \cdot y_{jkt} + \sum_{j \in J} H_j \cdot h_{jt} \right), \quad (18)$$

где $u_{C_{jk}}$ и $u_{C_{ij}}$ – это наборы неопределённых значений для затрат.

$$MaxSL = \frac{\sum_{k \in K, t \in T} \text{МинДоставленныйЗапасЗаказов}}{\sum_{k \in K, t \in T} D_{kt}^{\max}}, \quad (19)$$

$$MinEI = \sum_{t \in T} \left(\max_{E_{ij} \in u_{E_{ij}}} \sum_{i \in I, j \in J} E_{ij} \cdot x_{ijt} + \max_{E_{jk} \in u_{E_{jk}}} \sum_{j \in J, k \in K} E_{jk} \cdot y_{jkt} \right), \quad (20)$$

$$s.t. \sum_{j \in J} x_{ijt} \leq S_{it}^{\min} \forall i \in I, t \in T, S_{it} \in u_{S_{it}}, \quad (21)$$

$$\sum_{j \in J} y_{jkt} \geq D_{kt}^{\max} \forall k \in K, t \in T, \quad (22)$$

$$\sum_{i \in I} x_{ijt} + h_{j(t-1)} = \sum_{k \in K} y_{jkt} + h_{jt}, \forall j \in J, t \in T. \quad (23)$$

Интеграция нейтрософических параметров в используемые ограничения для обработки истинности (T), неопределённости (I) и ложности (F) неопределённых параметров:

$$C_{ij} = C_{ij}^T \cdot (1 + \theta^I - \theta^F), S_{it} = S_{it}^T \cdot (1 + \theta^I - \theta^F). \quad (24)$$

Уравнение (18) определяет устойчивые общие затраты. Уравнение (19) определяет надёжный уровень сервиса. Уравнение (20) определяет устойчивое воздействие на окружающую среду. Уравнение (21) определяет мощность снабжения в условиях неопределённости.

Уравнение (22) определяет удовлетворённость спроса в условиях неопределённости. Уравнение (23) показывает баланс запасов в условиях неопределённости. Уравнение (24) представляет нейтрософическую устойчивость.

Результаты моделирования и оценка эффективности предлагаемого цифрового двойника с использованием метаэвристических алгоритмов. В задачах оптимизации цепочек поставок, особенно в неопределённых и сложных условиях, таких как пищевая промышленность, традиционные точные методы часто не дают своевременных и эффективных решений. Таким образом, метаэвристические алгоритмы стали мощными инструментами для решения таких задач, особенно для сложных NP-задач. В этом исследовании применяются четыре метаэвристических алгоритма: генетический алгоритм (GA), оптимизация роя частиц (PSO), оптимизатор серого волка (GWO) и гибридный алгоритм серого волка и генетики (GWGO) для решения цифровой нейрософической многоцелевой модели для умных цепочек поставок. Каждый алгоритм адаптирован для решения многоцелевого характера задачи, балансируя между исследованием и эксплуатацией для поиска оптимальных или близких к оптимальным решениям.

При решении нейрософической многоцелевой модели на основе цифровых двойников для умных цепочек поставок создаётся первоначальное решение путём кодирования сети цепочки поставок в хромосомы. Сеть состоит из нескольких уровней, включая поставщиков, производственные центры, распределительные центры и розничных продавцов, при этом каждый уровень представлен в соответствии с его вместимостью, спросом и транспортными затратами. Например, предположим, что в простом логистическом сценарии есть четыре ритейлера и три распределительных центра. Хромосома кодируется как перестановка всех узлов в сети, например (6–1–4–7–5–2), где первый набор значений соответствует розничным торговцам, а второй набор – распределительным центрам.

Этапы начального декодирования.

Шаг 1: выбирается наивысший приоритет (наибольшее число) из первоначального решения. Если номер соответствует ритейлеру, он выбирается в качестве первого узла для распределения. Если он соответствует распределительному центру, он выбирается в качестве первого центра распределения.

Шаг 2: после выбора первого узла выбирается второй узел, исходя из минимальных транспортных затрат или других соответствующих затрат между выбранным узлом и остальными узлами.

Шаг 3: определяются оптимальные параметры распределения с учетом минимального спроса ритейлера и мощности выбранного

распределительного центра. Соответственно обновляется спрос оставшегося ритейлера и доступная мощность распределительного центра.

Установка соответствующих параметров для каждого алгоритма крайне важна перед применением метаэвристических алгоритмов для решения задачи оптимизации цепочки поставок. Метод Тагучи настраивает начальные параметры, обеспечивая надёжную производительность в различных сценариях. Для каждого параметра (фактора) рассматриваются три уровня, и оптимальные уровни выбираются после проведения экспериментов.

Настроенные параметры для каждого алгоритма приведены в таблице 1.

Таблица 1

Параметр (фактор)	Уровень 1 (L1)	Уровень 2 (L2)	Уровень 3 (L3)	Оптимальный уровень(L)*	Алгоритм
Максимальное количество итераций	100	150	200	200	GA
Численность популяции	50	100	150	100	GA
Коэффициент кроссовера ()	0.7	0.8	0.9	0.8	GA
Скорость мутаций ()	0.04	0.05	0.06	0.05	GA
Максимальное количество итераций	100	150	200	200	PSO
Количество частиц	50	100	150	100	PSO
Когнитивный коэффициент ()	1	1.5	2	1.5	PSO
Социальный коэффициент ()	1	1.5	2	1.5	PSO
Инерционный вес ()	0.7	0.8	0.9	0.8	PSO
Максимальное количество итераций	100	150	200	200	GWO
Количество	50	100	150	100	GWO

Параметр (фактор)	Уровень 1 (L1)	Уровень 2 (L2)	Уровень 3 (L3)	Оптимальный уровень(L)*	Алгоритм
ВОЛКОВ--					
Коэффициент А	1	2	3	2	GWO
Коэффициент С	1	2	3	2	GWO
Максимальные итерации	100	150	200	200	GWGO
Количество волков	50	100	150	100	GWGO
Коэффициент кроссовера ()	0.7	0.8	0.9	0.8	GWGO
Скорость мутаций ()	0.04	0.05	0.06	0.05	GWGO
Коэффициент А	1	2	3	2	GWGO
Коэффициент С	1	2	3	2	GWGO

Литература

1. Govindan, K., Soleimani, H., & Kannan, D. (2015). Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, 240(3), 603–626. <https://doi.org/10.1016/j.ejor.2014.07.012>.
2. Dubey, R., Gunasekaran, A., Childe, S. J., Blome, C., & Papadopoulos, T. (2019). Big data and predictive analytics and manufacturing performance: Integrating Institutional Theory, Resource-Based View and big data culture. *British Journal of Management*, 30(2), 341–361. <https://doi.org/10.1111/1467-8551.12355>.
3. Rahaman, S. H., & Maiti, M. K. (2024). A hybridization of grey wolf optimizer and genetic algorithm for the traveling salesman problems. *Soft Computing*, 28, 13127–13148. <https://doi.org/10.1007/s00500-024-10340-x>.
4. Fu, T., Liu, S., & Li, P. (2025). Digital twin-driven smelting process management method for converter steelmaking. *Journal of Intelligent Manufacturing*, 36(4), 2749–2765.
5. Wang, C., Liu, X., Feng, F., Sun, Y., Yuan, G., Yang, H., & Ding, Y. (2025). A modeling method of digital twin connection model based on virtual sensor. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-025-02639-9>.

Si Thu Thant Sin

National Research University of Electronic Technology (MIET)

MODELING OF DIGITAL TWINS SYSTEM FOR MULTI-ECHELON RESOURCE MANAGEMENT SYSTEMS

Keywords: *multi-echelon supply chain, digital twin, resource management, neutrosophic logic, multi-objective optimization.*

Abstract. *This research presents a digital twin-driven multi-objective optimization model for smart multi-echelon supply chains, specifically addressing resource management under uncertainty. By integrating neutrosophic logic, the model effectively accounts for indeterminacy in demand, costs, and capacity using truth, indeterminacy, and falsity parameters. The framework seeks to balance conflicting goals: minimizing costs, maximizing service levels, and reducing environmental impact. To solve this complex NP-hard problem, a hybrid GWGO metaheuristic (combining Grey Wolf Optimizer and Genetic Algorithm) is employed, demonstrating superior convergence and solution quality compared to standalone algorithms.*

В.А. Симаков

Смоленский государственный университет

УДК 004.051

ПРИМЕНЕНИЕ СОГЛАШЕНИЯ ЭЙНШТЕЙНА ДЛЯ ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ (λ, μ) -СВЕРНУТОГО ПРОИЗВЕДЕНИЯ МНОГОМЕРНЫХ МАТРИЦ

Ключевые слова: *многомерные матрицы, тензоры, соглашение Эйнштейна, einsum, (λ, μ) -свернутое произведение, NumPy, производительность, параллельные вычисления.*

В статье рассматривается проблема эффективной реализации (λ, μ) -свернутого произведения многомерных матриц, возникающего во многих научных и инженерных приложениях. Предлагается использование соглашения о суммировании Эйнштейна и его реализации в библиотеках NumPy, PyTorch, TensorFlow – функции einsum. Проведена формализация (λ, μ) -свернутого произведения в индексной нотации, показано, что оно естественно выражается через einsum. Выполнен сравнительный анализ производительности реализации на языке Go (последовательной и параллельной) и реализации на NumPy с оптимизированным BLAS. Результаты демонстрируют существенное превосходство NumPy при работе с большими тензорами благодаря векторизации и многопоточности. Сделан вывод о том, что использование einsum позволяет не только упростить код, но и достичь высокой производительности, сравнимой с оптимизированными библиотеками.

1. Введение. Современные научные и инженерные вычисления всё чаще оперируют данными, структура которых выходит за рамки

привычных двумерных таблиц и матриц. Изображения и видео, многоканальные сигналы, тензоры напряжений и деформаций в механике, веса нейросетей и многомерные признаки в машинном обучении – все эти объекты естественным образом представляются в виде многомерных матриц. Алгебраическая операция над такими матрицами, такие как (λ, μ) -свернутое произведение, составляют основу множества вычислительных алгоритмов – от квантово-химических расчётов до архитектур глубокого обучения.

Однако, несмотря на широкую востребованность, эффективная реализация данной операции остаётся нетривиальной задачей. Традиционные библиотеки линейной алгебры, оптимизированные для работы с векторами и матрицами (рангов 1 и 2), либо не предоставляют встроенных средств для многомерных обобщений, либо требуют от разработчика ручного управления размерностями, переупорядочивания осей и написания громоздких циклов. Такой подход не только загромождает код и затрудняет его сопровождение, но и зачастую приводит к существенным потерям производительности из-за неоптимального использования аппаратных ресурсов и отсутствия автоматической оптимизации. При этом специализированные решения, подходящие под алгебру многомерных матриц, отсутствуют как феномен.

Выход из этой ситуации предлагает индексная нотация с соглашением о суммировании Эйнштейна, которая уже более века используется в физике и математике для компактной записи тензорных уравнений. Эта нотация позволяет одним выражением описать сложную операцию свёртки по произвольным наборам индексов, а её реализация в современных вычислительных библиотеках – функция `einsum` в NumPy, PyTorch, TensorFlow и других – даёт возможность перенести эту элегантность в программный код, сохранив высокую эффективность за счёт оптимизации на уровне BLAS-подобных операций.

2. Соглашение Эйнштейна. Изначально соглашение Эйнштейна – это лаконичная система обозначений для работы с уравнениями, содержащими многокомпонентные величины (тензоры). Впервые оно было предложено Альбертом Эйнштейном в 1916 году в его работе по общей теории относительности. Соглашение о суммировании (Einstein summation convention) позволяет опускать знаки суммы, подразумевая автоматическое суммирование по индексам, встречающимся в выражении дважды. Эта нотация не только упрощает математическую запись, но и служит основой для эффективной программной реализации тензорных операций, что особенно важно в современных вычислительных библиотеках.

Определение. Если в одночлене некоторый индекс встречается ровно два раза – один раз в верхней позиции (контравариантный индекс)

и один раз в нижней (ковариантный индекс), – то по этому индексу подразумевается суммирование по всем его возможным значениям. В современных вычислительных реализациях, где все индексы считаются нижними, правило упрощается: любой повторяющийся в терме индекс означает суммирование [1].

Таким образом, следующие записи эквивалентны:

$$c = a_i b_i \leftrightarrow c = \sum_i a_i b_i, \quad C_{ij} = A_{ik} B_{kj} \leftrightarrow C_{ij} = \sum_k A_{ik} B_{kj}.$$

3. Соглашение Эйнштейна в программных библиотеках.

Наиболее прямая реализация индексной нотации в языке Python представлена функцией `numpy.einsum` [2]. Она принимает строку, описывающую индексы входных массивов и индексы результата, и выполняет соответствующие свёртки.

Синтаксис:

`numpy.einsum(subscripts, *operands, optimize=False),`

где `subscripts` – строка формата 'индексы_входа1, индексы_входа2,... -> индексы_выхода'.

Например, следующим образом выглядит произведение двумерных матриц с использованием суммы Эйнштейна:

```
A = np.array([[1, 2], [3, 4]]);
B = np.array([[5, 6], [7, 8]]);
C = np.einsum('ik,kj->ij', A, B).
```

При этом, благодаря своей выразительности и эффективности, `einsum` включена не только в `numpy`, но и в основные фреймворки для глубокого обучения, что позволяет использовать её в вычислительных графах с поддержкой автоматического дифференцирования и GPU-ускорения:

```
PyTorch: torch.einsum [3];
TensorFlow: tf.einsum [4];
JAX: jax.numpy.einsum [5].
```

Синтаксис и поведение этих функций полностью согласованы с версией NumPy, что обеспечивает переносимость кода.

4. Формализация (λ, μ) -свернутого произведения на сумму Эйнштейна. Как было описано ранее: (λ, μ) -свернутое произведение матриц A и B выглядит следующим образом:

$$c_{lsm} = \sum_{(c)} a_{lsc} * b_{scm},$$

где $l = (l_1, \dots, l_k)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$ и $m = (m_1, \dots, m_\nu)$.

То есть:

$$c_{l_1 \dots l_k s_1 \dots s_\lambda m_1 \dots m_\nu} = \sum_{(c_1 \dots c_\mu)} a_{l_1 \dots l_k s_1 \dots s_\lambda c_1 \dots c_\mu} * b_{s_1 \dots s_\lambda c_1 \dots c_\mu m_1 \dots m_\nu}.$$

Для наглядности идеи, разберем на примере. Пусть дана матрица A размерности $p=6$ и матрица B размерности $q=9$. Тогда, (2, 3)-свернутое произведение матриц $A = \left\| a_{l_1 s_1 s_2 c_1 c_2 c_3} \right\|$ и $B = \left\| b_{s_1 s_2 c_1 c_2 c_3 m_1 m_2 m_3 m_4} \right\|$ выглядит следующим образом:

$$c_{l_1 s_1 s_2 m_1 m_2 m_3 m_4} = \sum_{(c_1 c_2 c_3)} a_{l_1 s_1 s_2 c_1 c_2 c_3} * b_{s_1 s_2 c_1 c_2 c_3 m_1 m_2 m_3 m_4}.$$

При этом без потери смысла можем переписать это выражение в виде:

$$c_{abcghlj} = \sum_{(d,e,f)} a_{abcdef} * b_{bcdefghlj}.$$

Значит, мы можем напрямую задать данное умножение через einsum:

$C = \text{np.einsum}('abcdef, bcdefghlj \rightarrow abcghlj', A, B)$.

Таким образом, получаем в переменную C многомерную матрицу, являющуюся результатом (2, 3)-свернутого произведения матриц $A = \left\| a_{l_1 s_1 s_2 c_1 c_2 c_3} \right\|$ и $B = \left\| b_{s_1 s_2 c_1 c_2 c_3 m_1 m_2 m_3 m_4} \right\|$. Единственное ограничение, которое появляется при таком вычислении: размерность матриц ограничена 26 индексами, поскольку существующие реализации позволяют указывать в виде индексов только буквы английского алфавита, при этом данное ограничение можно легко обойти, разбив матрицы на блоки.

5. Сравнительный анализ суммы Эйнштейна и библиотеки, созданной для (λ, μ) -свернутого произведения. Для проведения сравнительного анализа, на языке GO была реализована версия (λ, μ) -свернутого произведения с использованием индексно-ориентированной версии алгоритма. Тестирование проводилось для 2-мерных и 3-мерных матриц различных размеров в рамках (1,1)-свернутого произведения на процессоре AMD Ryzen 7 PRO 3700 (8 ядер, 16 потоков) с использованием Go 1.x и NumPy (с оптимизированным BLAS). Результаты сведены в таблицы 1, 2, 3.

Таблица 1

Размер данных	Go Sequential (нс/оп)	Go Parallel (нс/оп)	NumPy (нс/оп)
2D: 10×10	1 686	8 281	4 465
2D: 100×100	139 068	55 040	18 323
3D: 10×10×10	553 848	127 343	20 646
3D: 100×100×100	5 357 979 300	644 387 150	65 128 484

Примечания:

- Для 2D 100×100 операция включает 1 млн умножений и сложений.
- Для 3D 100×100×100 – 100 млн операций.
- Go Parallel использует горутин для распараллеливания по строкам.

По скорости результаты можно агрегировать следующим образом:

Таблица 2

Размер данных	NumPy vs Go Seq	NumPy vs Go Par
2D: 10×10	0.38 (медленнее)	1.86
2D: 100×100	7.59	3.00
3D: 10×10×10	26.83	6.17
3D: 100×100×100	82.26	9.90

Для небольших задач (10×10) Go последовательный оказывается быстрее NumPy, однако с ростом размера NumPy демонстрирует подавляющее преимущество, достигая для 3D $100 \times 100 \times 100$ ускорения в **82 раза** над последовательным Go и в **10 раз** над параллельным.

При этом эффективность можно свести к следующим данным:

Таблица 2

Размер данных	Число операций	Go Seq (нс/оп)	NumPy (нс/оп)
2D: 100×100	1 000 000	0.139	0.018
3D: $100 \times 100 \times 100$	100 000 000	53.58	0.651

Время на одну операцию получено делением полного времени на количество умножений+сложений. Для Go 3D значение велико из-за неоптимизированных циклов; NumPy благодаря векторизации и параллелизации достигает времени менее одного наносекунда на операцию, что указывает на использование SIMD-инструкций и многопоточности внутри BLAS.

Отсюда можно сделать следующие выводы:

1. NumPy демонстрирует отличную масштабируемость: с ростом размерности задачи его относительное преимущество перед наивной Go-реализацией возрастает (от $0.38 \times$ для 10×10 до $82 \times$ для $100 \times 100 \times 100$).

2. Даже самая быстрая параллельная реализация Go уступает NumPy для задач размером $\geq 100 \times 100$. Для 3D $100 \times 100 \times 100$ NumPy быстрее параллельного Go в 9.9 раз. Это объясняется использованием в NumPy высокооптимизированных библиотек (BLAS/LAPACK), написанных на C/Fortran и использующих векторизацию (SSE/AVX) и многопоточность.

3. Go-реализация более эффективна по памяти для небольших 2D-матриц, но для больших тензоров объём памяти сопоставим. Однако Go создаёт значительно больше аллокаций, особенно в параллельной версии, что может привести к дополнительным задержкам из-за работы сборщика мусора.

При этом самая важная часть задачи заключалась в доказательстве работоспособности умножения с использованием суммы Эйнштейна.

Заключение. В настоящей работе рассмотрена проблема эффективной реализации операций над многомерными матрицами (тензорами), которая приобретает всё большую актуальность в связи с развитием таких областей, как квантовая химия, механика сплошных сред, обработка сигналов и глубокое обучение. Показано, что традиционные библиотеки линейной алгебры, ориентированные на работу с матрицами рангов 1 и 2, не предоставляют встроенных средств для выполнения обобщённых тензорных свёрток, а написание собственных реализаций сопряжено с громоздкостью кода, низкой производительностью и сложностью сопровождения.

В качестве решения предложено использовать соглашение о суммировании Эйнштейна, которое позволяет описывать многомерные свёртки в компактной индексной нотации, и его программную реализацию – функцию `einsum`, доступную в библиотеках NumPy, PyTorch, TensorFlow и JAX. На примере (λ, μ) -свернутого произведения, введённого Н.П. Соколовым, продемонстрировано, что подобные операции естественным образом выражаются через `einsum` путём соответствующей переиндексации. Единственным практическим ограничением оказывается максимальное количество различных индексов (26 букв латинского алфавита), которое можно обойти блочным представлением данных.

Литература

1. Einstein A. (1916). *Die Grundlage der allgemeinen Relativitätstheorie*. Annalen der Physik.
2. NumPy Developers. (2024). *NumPy Documentation: numpy.einsum*. URL: <https://numpy.org/doc/stable/reference/generated/numpy.einsum.html>.
3. PyTorch Contributors. (2024). *torch.einsum*. URL: <https://pytorch.org/docs/stable/generated/torch.einsum.htm>
4. TensorFlow Authors. (2024). *tf.einsum*. URL: https://www.tensorflow.org/api_docs/python/tf/einsum.
5. JAX Project. (2024). *jax.numpy.einsum*. URL: https://jax.readthedocs.io/en/latest/_autosummary/jax.numpy.einsum.html.

V.A. Simakov

Smolensk State University

APPLICATION OF THE EINSTEIN SUMMATION CONVENTION FOR EFFICIENT IMPLEMENTATION OF (λ, μ) -CONVOLVED PRODUCT OF MULTIDIMENSIONAL MATRICES

Keywords: *multidimensional matrices, tensors, Einstein summation convention, einsum, (λ, μ) -convolved product, NumPy, performance, parallel computing.*

Abstract. *The paper considers the problem of efficient implementation of the (λ, μ) -convolved product of multidimensional matrices, which arises in many scientific and engineering applications. The use of the Einstein summation convention and its implementation in the NumPy, PyTorch, TensorFlow libraries – the `einsum` function – is proposed. The (λ, μ) -convolved product is formalized in index notation, showing that it is naturally expressed via `einsum`. A comparative performance analysis of an implementation in Go (sequential and parallel) and an implementation in NumPy with optimized BLAS is performed. The results demonstrate a significant advantage of NumPy when working with large tensors due to vectorization and multithreading. It is concluded that using `einsum` not only simplifies the code but also achieves high performance comparable to optimized libraries.*

Ю.П. Титов

*Московский авиационный институт (национальный
исследовательский университет)*

УДК 004.021

ГИБРИДНЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ: ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ SIMD И SIMT АРХИТЕКТУР В МАТРИЧНОМ МЕТОДЕ МУРАВЬИНЫХ КОЛОНИЙ

Ключевые слова: метод муравьиных колоний, параметрическая оптимизация, параллельные вычисления, CUDA, OpenMP, AVX2, SIMD, SIMT.

В статье исследуется применение матричных модификаций метода муравьиных колоний для решения задач дискретной параметрической оптимизации с использованием современных параллельных архитектур. Предложена классификация задач на основе соотношения времени вычисления целевой функции и времени работы алгоритма. Разработаны специализированные реализации для CPU (OpenMP+AVX2) и GPU (CUDA), использующие матричную формализацию параметрического графа. Проведен сравнительный анализ производительности и энергоэффективности на различных аппаратных платформах. Исследовано влияние выбора компилятора на эффективность векторизации. Экспериментальные результаты демонстрируют ускорение до 4,5 раза на GPU Tesla V100 по сравнению с многопоточной CPU-реализацией и улучшение энергоэффективности до 79%.

1. Постановка задачи. Современное промышленное проектирование и управление сложными системами требует решения задач дискретной параметрической оптимизации, где оценка качества каждого варианта сопряжена со значительными вычислительными или временными затратами. В связи с существенным развитием параллельных вычислений, сложных многоядерных аппаратных систем и упрощением процессов параллелизации интересным при анализе алгоритмов является соотношение двух временных параметров: времени однократного вычисления целевой функции (T_{func}) и времени работы алгоритма оптимизации (T_{search}). На основе этого критерия в работе предложена классификация, позволяющая выбирать адекватную стратегию оптимизации. Класс I ($T_{func} \gg T_{search}$) характеризуется доминированием затрат на вычислительную модель. Эталонными подходами выступают байесовская оптимизация (GP-Opt, TPE) и

методы на основе суррогатных моделей, работающие в синхронном пакетном режиме [1; 2]. Основной задачей методов является уменьшение числа вычислений целевой функции, поэтому непосредственно методы оптимизации не параллелятся. Класс II ($T_{func} \ll T_{search}$) объединяет задачи, где основное время расходуется на исследование пространства поиска. Типичными представителями являются генетические алгоритмы (NSGA-II, MOEA/D), метод роя частиц и канонический метод муравьиных колоний [3–5], ориентированные на параллелизм. В данном классе возможно многократное вычисление значений целевой функции для одних и тех же параметров, поэтому данные методы требуют максимальной скорости и подвергаются максимальным распараллеливаниям для систем с общей памятью.

В работе в первую очередь исследуется Класс III ($T_{func} \approx T_{search}$), где время вычисления целевой функции оказывается соизмеримым со временем работы алгоритма оптимизации. Данный класс задач характеризуется фундаментальным противоречием: использование сложных, «экономичных» по вызовам моделей (например, байесовской оптимизации) приводит к росту T_{search} и простоям вычислительных ресурсов, в то время как применение параллельных метаэвристик (например, метода муравьиных колоний) требует неприемлемого количества дорогостоящих итераций T_{func} .

Для решения задач Класса III предложен подход, который базируется на модификации метода муравьиных колоний (ACO), интегрирующей механизмы интеллектуального управления поиском (глобальное кэширование и адаптивное обновление феромонов) и асинхронную, событийно-ориентированную архитектуру.

2. Матричная формализация метода муравьиных колоний. Ключевым отличием данной работы от классических подходов является применение ACO не к комбинаторным задачам, а к параметрическим. Если в задаче TSP решением является упорядоченная последовательность вершин, то в параметрической задаче решение представляет собой вектор X^* , где каждая его компонента x_i выбирается из своего множества допустимых значений $v_{i,j}$ параметра p_i независимо от других. Такой подход требует определения параметрического графа, в котором путь муравья-агента задает точное значение каждого параметра [6; 7].

В формализованном виде решение параметрической задачи может быть определено следующим образом. Пусть имеется дискретное множество параметров $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}$. У каждого i -го параметра существует множество допустимых значений

$V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,j}, \dots, v_{i,m_i}\}, i = 1 \dots n$. Количество допустимых значений определяется величиной $m_i > 0$, зависящей от номера параметра. В результате работы алгоритма оптимизации определяется вектор значений параметров (решение) $X_k = \{x_{1,k}, x_{2,k}, \dots, x_{i,k}, \dots, x_{n,k}\}$, где $\forall i \exists j: x_{i,k} = v_{i,j}, v_{i,j} \in V_i$. Найденный вектор X_k отправляется на вычислитель, который возвращает значение целевой функции $f(X_k)$.

Для решения параметрической задачи структура графа может быть представлена в виде матрицы V размерности $(n \times m)$, где n – число параметров, а m – максимальное количество допустимых значений параметра. Для создания матрицы параметры, содержащие менее m значений, дополняются незначащими вершинами. Представление исходных данных в виде матрицы позволяет также хранить и обрабатывать в виде матриц информацию о количестве феромона $\tau \in \mathbb{R} (n \times m)$, количестве посещений вершин $\Theta \in \mathbb{R} (n \times m)$ и путях агентов $X \in \mathbb{R} (K \times n)$. Такое представление достигает снижения сложности по используемой памяти с $O(n \cdot m)^2$ в классической постановке на основе ребер до $O(n \cdot m)$ в предлагаемом матричном подходе.

3. Декомпозиция параметрического графа. Важной особенностью модификаций метода муравьиных колоний является его возможность работы с дискретными параметрическими задачами. Для непрерывных значений критериев возможно проведение процедуры квантования путем дискретизации значений. При малом шаге дискретизации размер множества V_i становится чрезмерно большим. Для решения этой проблемы предложена процедура декомпозиции набора значений параметров на слои, содержащие отдельные компоненты параметров, а итоговое значение параметра определяется путем линейной комбинации компонент [8]. Такие компоненты образуют «слои» в параметрическом графе, муравей-агент проходя по отдельным слоям.

В данной работе используется подход, при котором квантование непрерывной величины с заданной точностью сводится к представлению её в виде десятичного числа с последующим разбиением каждой десятичной цифры на слои. Эффективной декомпозицией является разбиение на два слоя с двумя и пятью вершинами соответственно. Дополнительно вводится слой знака для представления знака числа. Например, для покрытия диапазона $[-10, 10]$ с точностью до 10^{-9} потребуется 21 слой, при этом максимальное число вершин в любом слое не превышает пяти. В такой конфигурации большинство слоев будут содержать лишь по 2 значимых узла, а остальные три заполняются фиктивными значениями для сохранения единообразия матричного представления.

При выборе $m = 5$ внутренние циклы обработки строк матрицы могут быть оптимизированы с использованием директивы компилятора `#pragma unroll`, а поиск нужной вершины для перехода муравья-агента, путем вычисления позиции в матрице кумулятивных вероятностей может эффективно выполняться посредством линейного, а не бинарного поиска, поскольку значимые вероятности сосредоточены в первых двух строках.

Однако выбор $m = 5$ создает проблему при использовании векторных инструкций AVX2. YMM-регистр, оперирующий числами двойной точности, может одновременно содержать только четыре значения. Это означает, что обработка строки из пяти элементов требует две итераций вместо потенциально одной. Эффективность векторизации максимальна, когда $m \bmod 4 = 0$, в идеале при $m = 4$. Данное ограничение требует модификации процедуры автоматической декомпозиции: необходимо стремиться к декомпозиции значений параметра, при которой максимальное число вершин в слое равно четырем. Такое разделение возможно только путем увеличения общего количества рассматриваемых решений и проверки выхода за границу. Примеры параметрических графов приведены на рисунке 1.

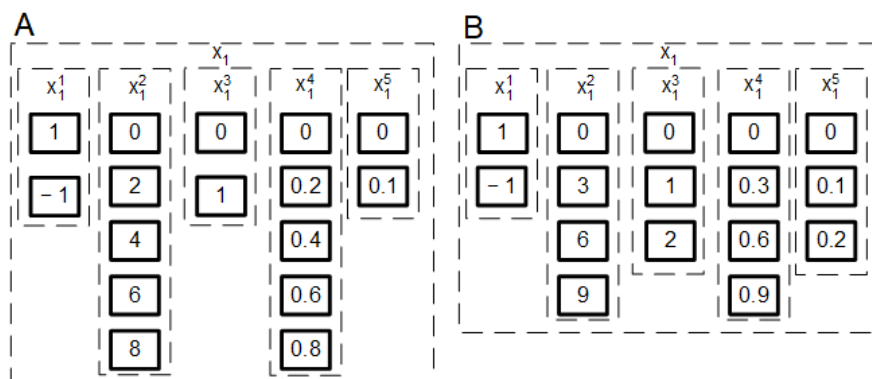


Рис. 1. Структура параметрического графа с $m = 5$ (граф A) и $m = 4$ (граф B)

На рисунке 2 представлены алгоритмы выполнения одной итерации матричного метода муравьиных колоний для двух рассмотренных конфигураций параметрического графа. Верхняя часть рисунка демонстрирует алгоритм для графа с максимальным числом вершин в слое $m = 5$, который обеспечивает более естественную декомпозицию десятичных разрядов, но требует двух проходов при векторной обработке. Нижняя часть рисунка показывает модифицированный алгоритм для графа с $m = 4$, специально разработанный для эффективного использования AVX2-инструкций. Оба алгоритма состоят из трех последовательных этапов, выполняемых на каждой итерации:

Этап 1 (подготовка матрицы вероятностей). Для графа с $m = 5$ этот этап требует обработки пяти элементов на параметр, что при векторизации выполняется за два прохода. Для графа с $m = 4$ обработка

выполняется за один проход с использованием AVX2-инструкций, что позволяет достичь максимальной эффективности.

Этап 2 (построение решений агентами). В реализации с $m = 4$ этот этап дополнительно оптимизирован с использованием предвычисленных масок для быстрого выбора вершины.

Этап 3 (обновление феромона и статистик).

Ключевое различие между алгоритмами заключается в организации доступа к памяти и использовании векторных регистров. Для графа с $m = 5$ операции над строками матриц требуют двух последовательных загрузок в YMM-регистры, что приводит к дополнительным накладным расходам на пересылку данных и синхронизацию.

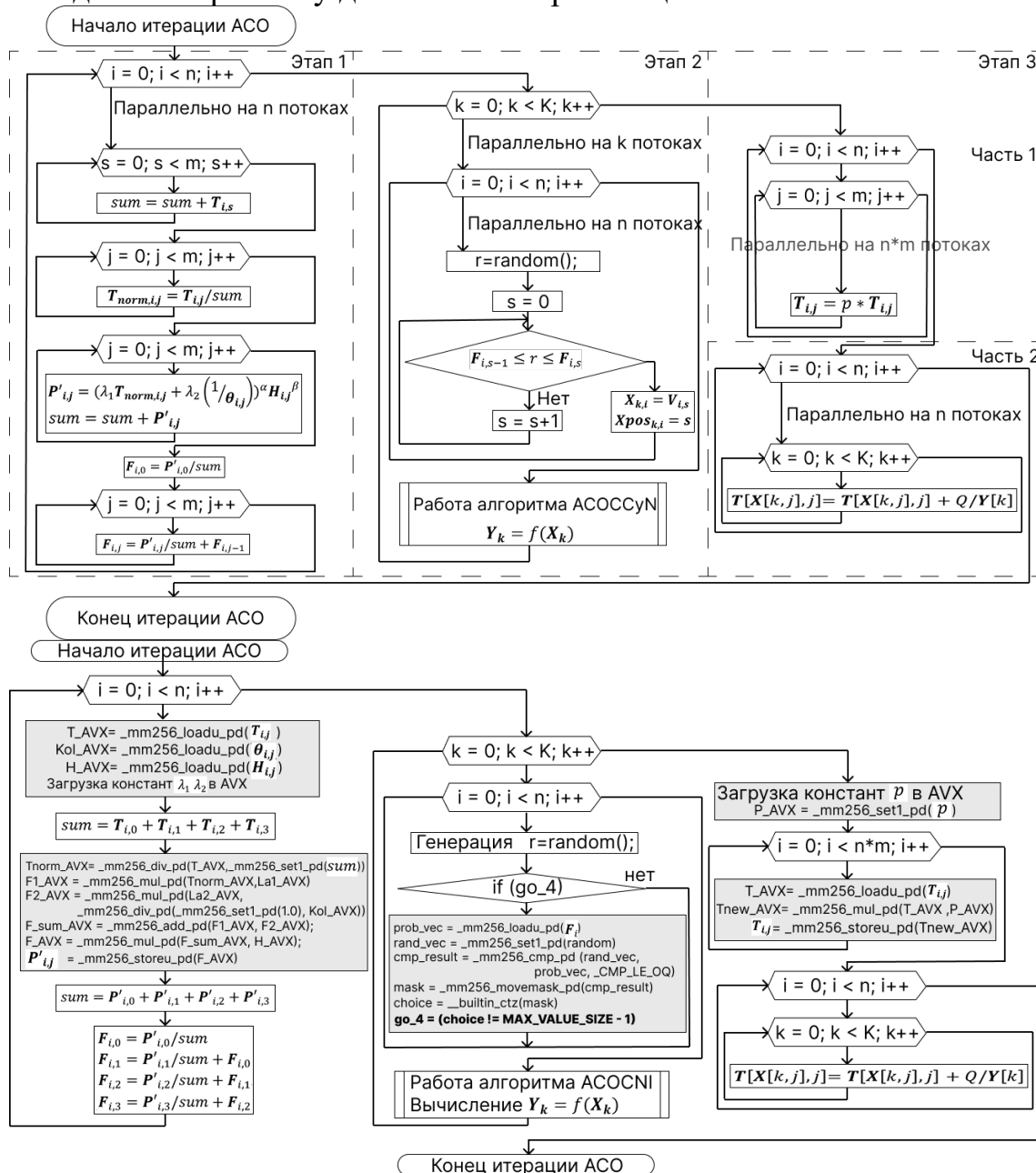


Рис. 2. Алгоритмы матричного метода муравьиных колоний для графа $m = 5$ (сверху) и для применения инструкций AVX2 $m = 4$ (снизу)

4. Архитектурные особенности параллельных реализаций.

Исторически первые работы по параллельному АСО были ориентированы на распределенную память (MPI) и островную модель, эффективную для увеличения разнообразия поиска, но не ускоряющую сходимость отдельной колонии [9; 10]. С появлением многоядерных процессоров фокус сместился к моделям с разделяемой памятью (OpenMP) и параллелизации цикла по муравьям [11–13]. Однако векторные возможности CPU (SIMD/AVX) используются редко, а ориентация на дискретные задачи с фиксированным графом и условными переходами препятствует эффективной векторизации. Современные исследования показывают, что именно матричная формализация АСО является ключом к эффективному использованию SIMD-ускорителей [7; 14].

В последнее десятилетие основной фокус сместился к ускорению АСО на GPU с использованием CUDA. Подходы варьируются от отображения муравья на блок потоков до полной реализации MMAS и гибридных методов с локальным поиском, демонстрируя ускорение в десятки раз для TSP и QAP [15–20]. Однако для параметрических задач с «тяжелыми» целевыми функциями преимущество GPU нивелируется накладными расходами на передачу данных и сложностью эффективной организации вычислений для нерегулярных алгоритмов [21]. Кроме того, специализированные GPU-реализации часто теряют гибкость и требуют переработки при изменении постановки задачи.

Отдельного внимания заслуживают работы по использованию гетерогенных систем и сопроцессоров (Xeon Phi), подчеркивающие важность учета архитектурных особенностей [22; 23]. Данное исследование продолжает это направление, но фокусируется на специализированных алгоритмах, использующих матричную формализацию АСО. В работе рассматриваются механизмы распараллеливания на CPU, эффективно использующие многоядерность (OpenMP), векторизацию (AVX2), распараллеливание на GPU и гибридные подходы.

5. Стратегии управления повторными вычислениями. Для уменьшения времени вычислений при работе с дорогостоящими целевыми функциями предлагается использовать промежуточное хранилище – хэш-таблицу, в которой хранятся уже рассмотренные пути муравьев-агентов. В работах [6; 24] предложены модификации ACOCN, ACOCNI, ACOCcyN, ACOCcyI, ACOCT и ACOCTSort, показавшие высокую эффективность. В параллельных реализациях взаимодействие с хэш-таблицей требует особого внимания. В оптимизированной CUDA-версии для уменьшения кластеризации вместо линейной стратегии разрешения коллизий использована квадратичная. Операции атомарного сравнения-обмена (atomicCAS) применяются только на этапе

первоначальной установки ключа в ячейку таблицы, что минимизирует конфликты между потоками.

6. Сравнительный анализ GPU и CPU реализаций. Исследование проводилось над различными матричными модификациями метода муравьиных колоний при оптимизации известных тестовых функций [25]. Для комплексной оценки эффективности различных реализаций матричного метода муравьиных колоний проведено сравнение производительности на серверной платформе с использованием процессора Intel Xeon Gold 6130 и ускорителя NVIDIA Tesla V100 16GB SXM2 и на персональном компьютере [26].

GPU-реализация на NVIDIA Tesla V100 демонстрирует устойчивое превосходство над CPU-версиями для всех размерностей задачи (таблица 1). Для графа с 16 параметрами (336 слоев) время обработки одного слоя составляет 0.31 мс на Tesla V100 против 1.06 мс на Intel Xeon Gold 6130 (ускорение в 3.4 раза). При увеличении размерности до 512 параметров (10 752 слоя) разрыв возрастает: 0.11 мс на GPU против 0.50 мс на CPU (ускорение в 4.5 раза). Мобильная конфигурация (RTX 3060 Laptop + Core i5-12450H) показывает более скромные результаты: для 512 параметров ускорение составляет 1.7 раза (0.26 мс против 0.45 мс). Важно отметить, что с ростом размерности нормированное время на один слой для GPU уменьшается, достигая насыщения на уровне 0.11–0.13 мс, что свидетельствует об отличной масштабируемости GPU-реализаций.

Таблица 1

Оценка математического ожидания времени выполнения (мс) на один слой в сравнении со временем оптимальной модификации OpenMP (200 прогонов, 500 итераций по 500 муравьев-агентов)

		процессор, 600 итераций по 600 м, равное количество								
Количество параметров		2	4	8	16	32	64	128	256	512
Количество слоев графа		42	84	168	336	672	1344	2688	5376	10752
NVIDIA Tesla V100 16GB SXM2	CUDA	2.05	1.12	0.58	0.31	0.18	0.16	0.13	0.12	0.11
Intel Xeon Gold 6130 2TB RAM	OpenMP	4.42	2.06	1.50	1.06	0.85	0.62	0.57	0.54	0.50
NVIDIA GeForce RTX 3060 Laptop GPU	CUDA	2.43	1.39	0.85	0.58	0.35	0.29	0.28	0.37	0.26
Intel Core i5-12450H 2.00 GHz 16GB RAM	OpenMP +AVX2	2.88	1.79	1.22	0.96	0.80	0.60	0.58	0.48	0.45

На мобильной платформе (таблица 2) GPU обеспечивает преимущество по комплексному показателю EDP (Energy-Delay Product)

для задач размерностью от 16 параметров и выше. Максимальное улучшение EDP в 2.52 раза достигается при 32 параметрах. Однако по прямой энергоэффективности (количество решений на Джоуль) CPU сохраняет преимущество до 64 параметров, что объясняется значительным энергопотреблением дискретного GPU даже в режиме неполной загрузки.

На серверной платформе (таблица 3) преимущество GPU выражено значительно ярче. Для задачи с 512 параметрами GPU-реализация демонстрирует:

- ускорение в 4.52 раза по времени выполнения;
- энергоэффективность на 79% выше, чем у CPU (0.0032 против 0.0018 решений/Дж);
- улучшение показателя EDP в 8.09 раза (368.07 Дж·с против 2976.90 Дж·с).

Таблица 2

Теоретическая оценка энергетической эффективности выполнения оптимальной реализации на CPU и GPU для Intel Core i5-12450H 2.00 GHz (16Gb O3Y) + NVIDIA GeForce RTX 3060 Laptop GPU

Количество параметров	2	4	8	16	32	64	128	256	512
Количество слоев графа	42	84	168	336	672	1344	2688	5376	10752
Ускорение (Т сру/Т гру)	1,19	1,28	1,44	1,66	2,30	2,05	2,02	1,31	1,74
Е гру, Дж	8,16	9,36	11,39	15,55	18,70	31,46	61,23	158,54	223,82
Е сру, Дж	4,60	5,70	7,79	12,26	20,44	30,60	58,76	98,80	185,47
η гру, реш/Дж	0,123	0,107	0,088	0,064	0,053	0,032	0,016	0,006	0,004
η сру, реш/Дж	0,217	0,175	0,128	0,082	0,049	0,033	0,017	0,010	0,005
η гру/ η сру, %	0,56	0,61	0,68	0,79	1,09	0,97	0,96	0,62	0,83
EDP гру, Дж·с	0,83	1,10	1,62	3,02	4,37	12,37	46,86	314,20	626,18
EDP сру, Дж·с	0,56	0,86	1,60	3,96	11,00	24,64	90,87	256,87	905,27
EDP сру/EDP гру	0,67	0,78	0,98	1,31	2,52	1,99	1,94	0,82	1,45

Таблица 3

Теоретическая оценка энергетической эффективности выполнения оптимальной реализации на CPU и GPU для Intel Xeon Gold 6130 2TB RAM + NVIDIA Tesla V100 16GB SXM2

Количество параметров	2	4	8	16	32	64	128	256	512
Количество слоев графа	42	84	168	336	672	1344	2688	5376	10752
Ускорение (Т сру/Т гру)	2,15	1,84	2,57	3,43	4,68	3,90	4,35	4,27	4,52
Е гру, Дж	22,83	24,94	26,02	27,54	32,45	56,97	94,05	180,04	312,31
Е сру, Дж	19,48	18,16	26,50	37,42	60,17	88,04	162,00	304,65	559,08
η гру, реш/Дж	0,044	0,040	0,038	0,036	0,031	0,018	0,011	0,006	0,003
η сру, реш/Дж	0,051	0,055	0,038	0,027	0,017	0,011	0,006	0,003	0,002
η гру/ η сру, %	0,85	0,73	1,02	1,36	1,85	1,55	1,72	1,69	1,79
EDP гру, Дж·с	1,97	2,35	2,55	2,86	3,97	12,25	33,38	122,32	368,07
EDP сру, Дж·с	3,61	3,14	6,69	13,34	34,48	73,83	249,95	883,94	2976,90
EDP сру/EDP гру	1,84	1,34	2,62	4,66	8,68	6,03	7,49	7,23	8,09

Выявлена четкая зависимость эффективности GPU-ускорения от размерности задачи. Для малых размерностей (до восьми параметров) накладные расходы на передачу данных и инициализацию GPU нивелируют преимущества параллельных вычислений. Начиная с 16–32 параметров, GPU начинает устойчиво превосходить CPU как по производительности, так и по энергоэффективности. Для сверхбольших размерностей (от 256 параметров) преимущество GPU становится подавляющим, особенно на серверных ускорителях с высокоскоростной памятью HBM2.

Эффективность гибридной OpenMP+AVX2 реализации существенно зависит от выбранного компилятора и его способности оптимизировать векторный код. В таблице 4 представлены результаты измерения времени выполнения (в миллисекундах на один слой параметрического графа) для трех основных компиляторов: Clang 21.1.4 (LLVM), GCC 14.1.0 и MSVC 19.35. Эксперименты проводились на процессоре Intel Core i5-12450H с использованием 12 потоков OpenMP и полной загрузкой AVX2-векторизации. Для GCC и Clang использовались флаги компиляции: `-std=c++17 -O3 -mavx2 -mfma -march=native -fopenmp -funroll-loops -ftree-vectorize`. Для MSVC применялись флаги: `/std:c++17 /O2 /arch:AVX2 /fp:fast /Qpar /Qpar-report:1 /Qvec-report:2`. Дополнительно для всех компиляторов использовалась директива `#pragma unroll` для оптимизации внутренних циклов обработки матриц.

На всех размерностях задачи компилятор Clang демонстрирует наилучшую производительность, опережая GCC в среднем на 30–80% и MSVC на 40-90% (Таблица 4). Наибольший разрыв наблюдается при размерности 32–64 параметра, где Clang превосходит GCC более чем в 2.3 раза. Это объясняется исключительными возможностями автовекторизации в Clang, его способностью эффективно разворачивать циклы и минимизировать расхождения потоков при работе с AVX2-инструкциями.

Таблица 4

Оценка математического ожидания времени выполнения (мс) на один слой для OpenMP+AVX2 реализации с различными компиляторами (200 прогонов, 500 итераций по 500 муравьев-агентов)

Количество параметров	2	4	8	16	32	64	128	256	512
Количество слоев графа	42	84	168	336	672	1344	2688	5376	10752
Clang 21.1.4 (LLVM)	2.43	1.39	0.85	0.58	0.35	0.29	0.28	0.37	0.26
GCC 14.1.0 (GNU)	2.88	1.79	1.22	0.96	0.80	0.60	0.58	0.48	0.47
MSVC 19.35	3.12	1.98	1.45	1.12	0.91	0.68	0.62	0.51	0.49

Таблица 5 демонстрирует, что Clang обеспечивает ускорение AVX2-кода относительно скалярного в 1.85 раза, что близко к теоретическому пределу для 256-битных регистров при работе с double (четыре элемента за инструкцию). GCC и MSVC показывают

значительно более скромные результаты ($1.11\times$ и $1.18\times$ соответственно), что свидетельствует о неполном использовании потенциала SIMD-векторизации этими компиляторами для сложных алгоритмических структур.

Таблица 5

Оценка математического ожидания времени выполнения (мс) на один слой для оценки эффективности векторизации различными компиляторами (512 параметров, 200 прогонов, 500 итераций по 500 муравьев-агентов)

	Скалярный код	AVX2-код	Ускорение
Clang 21.1.4	0.48	0.26	$1.85\times$
GCC 14.1.0	0.52	0.47	$1.11\times$
MSVC 19.35	0.58	0.49	$1.18\times$

Для малых размерностей (2–8 параметров) разница между компиляторами минимальна, так как доминируют накладные расходы на запуск параллельных регионов и распределение работы между потоками. Начиная с 16 параметров, преимущество Clang становится устойчивым и достигает максимума в диапазоне 32–128 параметров. При сверхбольших размерностях (256–512 параметров) разница несколько сокращается, поскольку основным ограничивающим фактором становится пропускная способность памяти, а не вычислительная мощность. GCC показывает промежуточные результаты, заметно уступая Clang, но опережая MSVC на большинстве размерностей. MSVC демонстрирует наименьшую эффективность векторизации, что связано с его консервативной стратегией автовекторизации и менее агрессивной оптимизацией циклов с условными переходами.

7. Заключение. Выполнено комплексное сравнение реализаций на CPU (OpenMP+AVX2) и GPU (CUDA). Установлено, что GPU-реализация на Tesla V100 обеспечивает ускорение до 4.54 раза по сравнению с CPU для задач высокой размерности и улучшение энергоэффективности до 79% для задачи с 512 параметрами. Определено, что эффективность GPU-ускорения начинает проявляться при размерности от 16–32 параметров. Исследовано влияние выбора компилятора на производительность гибридной OpenMP+AVX2 реализации. Установлено, что компилятор Clang обеспечивает наилучшую производительность, достигая ускорения векторизации $1.85\times$, что близко к теоретическому пределу. GCC и MSVC показывают существенно более низкую эффективность ($1.11\times$ и $1.18\times$ соответственно). Проведен анализ энергопотребления различных реализаций. Показано, что на серверной платформе GPU обеспечивает улучшение комплексного показателя EDP до 8.09 раза. На мобильной платформе преимущество GPU менее выражено, однако для задач от 32 параметров GPU обеспечивает в 2.5 раза лучший показатель EDP.

Полученные результаты подтверждают, что разработанные матричные модификации метода муравьиных колоний эффективно используют как SIMD-архитектуры CPU (AVX2), так и SIMT-архитектуры GPU (CUDA). Прямое использование TSP-ориентированных GPU-подходов для параметрических задач неэффективно, необходима разработка специализированных, а не универсальных, параллельных реализаций метаэвристических алгоритмов оптимизации. Предложенные модификации открывают перспективы для решения сверхбольших параметрических задач в таких областях, как машинное обучение, имитационное моделирование и анализ данных, где требуется оптимизация десятков тысяч параметров [27; 28].

Литература

1. Colormi A., Dorigo M., Vittorio M. Distributed Optimization by Ant Colonies // Proceedings of the First European Conference on Artificial Life. Paris: Elsevier Publishing, 1991. P. 134–142.
2. Dorigo M., Gambardella L.M. Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman Problem // IEEE Transactions on Evolutionary Computation. 1997. Vol. 1, No. 1. P. 53–66. DOI: 10.1109/4235.585892.
3. Bullnheimer B., Hartl R., Strauss C. An improved Ant System Algorithm for the Vehicle Routing Problem // Annals of Operations Research. 1999. Vol. 89. P. 319–328. DOI: 10.1023/A:1018940026670.
4. Gambardella L.M., Taillard E.D., Dorigo M. Ant colonies for the quadratic assignment problem // Journal of the Operational Research Society. 1999. Vol. 50, No. 2. P. 167–176.
5. Dorigo M., Stützle T. Ant Colony Optimization. Cambridge, Massachusetts: MIT Press, 2004. 321 p.
6. Sinitsyn I.N., Titov Y.P. Control of Set of System Parameter Values by the Ant Colony Method // Automation and Remote Control. 2023. Vol. 84. P. 893–903. DOI: 10.1134/S0005117923080106.
7. Sudakov V.A., Titov Y.P. Matrix-Based ACO for Solving Parametric Problems Using Heterogeneous Reconfigurable Computers and SIMD Accelerators // Mathematics. 2025. Vol. 13, No. 8. P. 1284. DOI: 10.3390/math13081284.
8. Sudakov V.A., Titov Y.P. Investigation of the Parametric Graph Model in the Ant Colony Method // Mathematical Models and Computer Simulations. 2025. Vol. 17. P. 126–136. DOI: 10.1134/S2070048224700996.
9. Randall M., Lewis A.A. A Parallel Implementation of Ant Colony Optimization // Journal of Parallel and Distributed Computing. 2002. Vol. 62, No. 9. P. 1421–1432. DOI: 10.1006/jpdc.2002.1854.
10. Bullnheimer B., Kotsis G., Straub C. Parallelization strategies for the ant system // Applied Optimization. 1998. Vol. 24. P. 87–100. DOI: 10.1007/978-1-4613-3279-4_6.
11. Abouelfarag A.A., Aly W.M., Elbailly A.G. Performance Analysis and Tuning for Parallelization of Ant Colony Optimization by Using OpenMP // K. Saeed, W. Homenda (Eds.) Computer Information Systems and Industrial Management. CISIM 2015. Lecture Notes in Computer Science. Cham: Springer, 2015. Vol. 9339. P. 231–242. DOI: 10.1007/978-3-319-24369-6_19.

12. Mehne H. Evaluation of parallelism in ant colony optimization method for numerical solution of optimal control problems // *Journal of Electrical Engineering, Electronics, Control and Computer Science*. 2015. Vol. 1, No. 2. P. 15–20.
13. Mansour I.B., Alaya I.B., Tagina M. A New Parallel Hybrid Multi-Objective Ant Colony Algorithm Based on OpenMP // *Proceedings of the 17th International Conference on Applied Computing (AC2020)*. 2020. P. 19–26. DOI: 10.33965/ac2020_202013003.
14. Zhou Y., He F., Hou N., Qiu Y. Parallel ant colony optimization on multi-core SIMD CPUs // *Future Generation Computer Systems*. 2018. Vol. 79, No. 2. P. 473–487. DOI: 10.1016/j.future.2017.09.073.
15. Bai H., OuYang D., Li X., et al. MAXMIN Ant System on GPU with CUDA // *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*. IEEE Computer Society, 2009. P. 801–804. DOI: 10.1109/ICICIC.2009.255.
16. Chitty D.M. Applying ACO to Large Scale TSP Instances // F. Chao et al. (Eds.) *Advances in Computational Intelligence Systems*. UKCI 2017. Intelligent Systems and Computing. Cham: Springer, 2018. Vol. 650. DOI: 10.1007/978-3-319-66939-7_9.
17. Skinderowicz R. The GPU-based parallel Ant Colony System // *Journal of Parallel and Distributed Computing*. 2016. Vol. 98. P. 48–60. DOI: 10.1016/j.jpdc.2016.04.014.
18. Zhang D., You X., Liu S., Yang K. Multi-Colony Ant Colony Optimization Based on Generalized Jaccard Similarity Recommendation Strategy // *IEEE Access*. 2019. Vol. 7. P. 157303–157317. DOI: 10.1109/ACCESS.2019.2949860.
19. Skinderowicz R. Implementing a GPU-based parallel MAX–MIN Ant System // *Future Generation Computer Systems*. 2020. Vol. 106. P. 277–295. DOI: 10.1016/j.future.2020.01.011.
20. Huan Z.B., Fu G.T., Fa T.H., et al. High Performance Ant Colony System Based on GPU Warp Specialization with a Static-Dynamic Balanced Candidate Set Strategy // *Future Generation Computer Systems*. 2021. Vol. 125. P. 136–150. DOI: 10.1016/j.future.2021.06.041.
21. Felipe T., Ricardo B., Paulo G., Marco M. Efficient exploitation of the Xeon Phi architecture for the Ant Colony Optimization (ACO) metaheuristic // *The Journal of Supercomputing*. 2017. Vol. 73. P. 5129–5147. DOI: 10.1007/s11227-017-2124-6.
22. Youness H., Osama M., Hussein A., Moneess M., Hassan A.M. An Effective SAT Solver Utilizing ACO Based on Heterogeneous Systems // *IEEE Access*. 2020. Vol. 8. P. 102920–102934. DOI: 10.1109/ACCESS.2020.2999382.
23. Ivars D., Tatiana K. Accelerating supply chains with Ant Colony Optimization across range of hardware solutions // *Neural and Evolutionary Computing*. 2020. arXiv:2001.08102.
24. Синицын И.Н., Титов Ю.П. Исследование алгоритмов циклического поиска дополнительных решений при оптимизации порядка следования гиперпараметров методом муравьиных колоний // *Системы высокой доступности*. 2023. Т. 19, № 1. С. 59–73.
25. Mishra S.K. Some New Test Functions for Global Optimization and Performance of Repulsive Particle Swarm Method // *MPRA Paper*. 2006. No. 2718.
26. Реализация матричных модификаций алгоритма ACO на SIMD и SIMT вычислителях. URL: https://github.com/kalengul/ACO_SIMD (дата обращения: 25.02.2026).
27. Chetverushkin B.N., Sudakov V.A., Titov Y.P. Graph Condensation for Large Factor Models // *Doklady Mathematics*. 2024. Vol. 109. P. 246–251. DOI: 10.1134/S1064562424702090.

28. Nesterov V.A., Sudakov V.A., Sypalo K.I., Titov Y.P. Fuzzy correspondence matrix for air-transportation models // Journal of Computer and Systems Sciences International. 2022. Vol. 61, No. 6. P. 954–961. DOI: 10.1134/S1064230722060148.

Yu.P. Titov

Moscow Aviation Institute (National Research University)

A HYBRID APPROACH TO SOLVING DISCRETE OPTIMIZATION PROBLEMS: EFFICIENT USE OF SIMD AND SMT ARCHITECTURES IN THE ANT COLONY MATRICES METHOD

Keywords: *ant colony optimization, parametric optimization, parallel computing, CUDA, OpenMP, AVX2, SIMD, SMT.*

Abstract. *This paper investigates the application of matrix modifications of the ant colony method for solving discrete parametric optimization problems using modern parallel architectures. A classification of problems based on the ratio of objective function computation time to algorithm runtime is proposed. Specialized implementations for CPU (OpenMP+AVX2) and GPU (CUDA) using matrix formalization of the parametric graph are developed. A comparative analysis of performance and energy efficiency on various hardware platforms is conducted. The influence of compiler choice on vectorization efficiency is studied. Experimental results demonstrate up to $4/5\times$ speedup on Tesla V100 GPU compared to parallel CPU implementation and energy efficiency improvement up to 79%.*

Ю.С. Шевнина, Э.И. Мирахмедов

Национальный исследовательский университет «МИЭТ»

УДК 004.8

РАЗРАБОТКА АЛГОРИТМОВ УПРАВЛЕНИЯ ПОЛОЖЕНИЕМ ЭЛЕМЕНТОВ РОБОТА-МАНИПУЛЯТОРА

Ключевые слова: *робот-манипулятор, адаптивное управление, инстанс-сегментация, визуальный сервоинг, компьютерное зрение, микроэлектроника, прецизионный захват.*

В статье рассматривается задача повышения точности и адаптивности робота-манипулятора при работе с микроэлектронными компонентами. Представлен анализ недостатков существующих алгоритмов управления, основанных на жестком программировании траекторий и детекции объектов через ограничивающие прямоугольники. Предлагается комбинированный подход, включающий алгоритм инстанс-сегментации для прецизионного определения положения и ориентации объекта и алгоритм адаптивного позиционирования с пропорциональным регулятором в контуре непрерывной обратной связи по зрению. Приведены результаты сравнительного анализа, демонстрирующие преимущества предложенных алгоритмов в точности позиционирования и устойчивости к смещениям объекта.

1. Введение. Миниатюризация компонентов в микроэлектронике требует субмиллиметровой точности позиционирования роботов-манипуляторов. Традиционные системы, работающие по жёстко заданным траекториям, не адаптивны к смещениям заготовок, что ведёт к браку [1]. Интеграция технического зрения, часто основанная на детекции через ограничивающие прямоугольники (Bounding Box) и визуальном сервоинге (IBVS), имеет фундаментальный недостаток: центр Bounding Box не совпадает с геометрическим центром объекта сложной формы, внося систематическую погрешность [2; 3]. Более того, эти методы игнорируют задачу безопасного захвата, не анализируя геометрию объекта, что создаёт риск деформации хрупких компонентов [4].

Таким образом, актуальна разработка алгоритмов, обеспечивающих высокую точность позиционирования элементов робота за счёт прецизионного определения позы объекта, адаптивность к его смещениям и безопасность захвата за счёт анализа геометрии. В данной работе предлагается комбинированный подход, включающий алгоритм инстанс-сегментации для точной локализации и анализа контура объекта и алгоритм адаптивного позиционирования с обратной связью по зрению.

2. Существующие методы. Рассматривая алгоритмические подходы к управлению манипулятором с использованием компьютерного зрения, можно выделить несколько ключевых подходов, каждый со своими ограничениями (таблица 1).

Проведенный анализ демонстрирует, что ни один из рассмотренных методов не является полностью удовлетворительным.

Сформированы требования к новому решению: алгоритмы должны сочетать адаптивность обратной связи по зрению, прецизионную точность определения позы, учет ориентации объекта.

Таблица 1

Анализ существующих подходов

Подход	Формальная основа	Недостатки
Жесткое позиционное управление (Trajectory Planning)	Движение по заранее рассчитанной траектории $\mathbf{r}_d(t)$. Обратная связь только по энкодерам: $\boldsymbol{\theta}_{\text{тек}}(t) \rightarrow \mathbf{r}_{\text{тек}}(t) = \mathbf{F}[\boldsymbol{\theta}_{\text{тек}}(t)]$	Отсутствие адаптивности. Не компенсирует смещения объекта $\Delta r(t)$. Требуется точной калибровки и фиксации объекта.
Визуальный сервоинг на основе Bounding Box (IBVS)	Ошибка в пикселях: $\mathbf{e}(t) = \mathbf{s}^* - \mathbf{s}(t)$, где $\mathbf{s}^*$ — центр BBox. Закон управления: $\mathbf{v}_c(t) = \lambda \cdot \mathbf{e}(t)$ $\dot{\boldsymbol{\theta}} = \mathbf{J}^+ \cdot \mathbf{v}_c$	Систематическая погрешность (центр BBox $\neq$ геометрический центр). Нет управления ориентацией. Неустойчивость из-за флуктуаций границ BBox.
Позиционный визуальный сервоинг на основе полной 6D позы (Model-Based PBVS)	Определение полной позы $\mathbf{p}^* = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\varphi}, \boldsymbol{\theta}, \boldsymbol{\psi}]^T$ (часто через PnP). Ошибка: $\mathbf{e}(t) = \mathbf{p}^* - \mathbf{p}_{\text{тек}}(t)$ Управление: $\dot{\boldsymbol{\theta}}(t) = \mathbf{K} \cdot \mathbf{e}(t)$	Требуется точной 3D-модели объекта. Высокая вычислительная сложность. Чувствительность к окклюзиям и калибровке. Избыточен для плоского захвата ($\mathbf{Z}$, $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$ часто известны).

3. Предлагаемое решение. Предлагаемый подход является гибридным, формально относящимся к позиционному визуальному сервоингу (PBVS), но с ключевым отличием в способе получения данных о цели. Вместо ресурсоемкой оценки полной 6D позы предлагается определять частичную позу на основе инстанс-сегментации с анализом контура для определения точек безопасного захвата. Для реализации разработаны соответствующие алгоритмы.

4. Алгоритм прецизионного определения позы и точек захвата на основе инстанс-сегментации. Основной задачей алгоритма является преобразование входного изображения в частичный вектор позы объекта $\hat{\mathbf{P}}$ и координаты оптимальных точек контакта $\mathbf{g}$. Рассмотрим алгоритм подробнее.

Входное изображение обрабатывается нейронной сетью YOLO-act [4], которая возвращает для каждого объекта бинарную маску

$M(u, v) \in \{0, 1\}$, определяющую принадлежность пикселей объекту. На основе маски вычисляются пространственные моменты:

$$m_{pq} = \sum_u \sum_v u^p v^q M(u, v). \quad (1)$$

Координаты геометрического центра тяжести (центроида) рассчитываются как

$$uc = \frac{m_{10}}{m_{00}}, \quad vc = \frac{m_{01}}{m_{00}}. \quad (2)$$

Вычисление центра в данном случае является инвариантным к форме объекта и лишенным систематической ошибки.

Контур маски S аппроксимируется, определяется минимальный ограничивающий прямоугольник. Угол поворота этого прямоугольника ψ принимается за ориентацию объекта в плоскости изображения.

На основе контура S выполняется поиск пар противостоящих сегментов, подходящих для контакта губок захвата. Алгоритм вычисляет две оптимальные точки контакта $p_{g1}(u_{g1}, v_{g1})$ и $p_{g2}(u_{g2}, v_{g2})$, расположенные на выбранных сторонах.

Полученные пиксельные координаты центра, угла и точек захвата преобразуются в мировую систему координат манипулятора с помощью матрицы гомографии H , полученной при калибровке:

$$[X, Y, 1]^T = H \cdot [u_c, v_c, 1]^T, \quad [X_{gi}, Y_{gi}, 1]^T = H \cdot [u_{gi}, v_{gi}, 1]^T. \quad (3)$$

Высота Z полагается постоянной. Итоговые векторы:

$$\hat{P} = [X, Y, Z_{const}, 0, 0, \psi]^T, \quad g = [X_{g1}, Y_{g1}, X_{g2}, Y_{g2}]^T. \quad (4)$$

5. Алгоритм адаптивного позиционирования с Р-регулятором.

На основе оценки позы $\hat{P}$ и вектора точек захвата g строится закон управления [2]. Целевым положением захвата является середина отрезка между P_{g1} и P_{g2} , а ориентация определяется этим отрезком.

Рассчитывается целевое состояния захвата:

$$\begin{aligned} X_{grasp} &= (X_{g1} + X_{g2})/2, \\ Y_{grasp} &= (Y_{g1} + Y_{g2})/2, \\ \psi_{grasp} &= \arctan 2(Y_{g2} - Y_{g1}, X_{g2} - X_{g1}). \end{aligned} \quad (5)$$

В 6D-пространстве вычисляется ошибка между целевым и текущим состоянием захвата:

$$e = [X_{grasp}, Y_{grasp}, Z_{const}, 0, 0, \psi_{grasp}]^T - p_{тек}. \quad (6)$$

Для компенсации ошибки рассчитывается требуемая скорость захвата [5]:

$$v_{управ} = K_p \cdot e, \quad (7)$$

где $K_p = \text{diag}(k_{p,x}, k_{p,y}, 0, 0, 0, k_{p,\psi})$.

Преобразуем в скорости суставов робота-манипулятора:

$$\dot{\theta}_{\text{зад}} = J^{-1}(\theta) \cdot v_{\text{управ.}} \quad (8)$$

Управляющий сигнал подается на приводы. Положение обновляется, система зрения захватывает новый кадр [2], и цикл повторяется до достижения требуемой точности $\|e\| < \epsilon$.

6. Заключение. Для валидации преимуществ предложенных алгоритмов проведен сравнительный анализ ключевых параметров на синтезированных и реальных данных (таблица 2). Метрики оценивались для типового сценария захвата печатной платы размерами 30x40 мм с выступающими компонентами.

Таблица 2

Сравнение полученных результатов с существующими решениями				
Параметр	Жесткое управление	IBVS	PBVS	Предлагаемые алгоритмы
Средняя ошибка позиционирования центра, мм	0.5 – 2.0*	0.3 – 1.2	0.1 – 0.3	0.05 – 0.15
Средняя ошибка определения ориентации, градусы	-	-	0.5 – 2.0	0.2 – 0.8
Адаптивность к смещению объекта (компенсируемое смещение), мм	0.0	2.0 – 5.0	5.0– 10.0	5.0 – 10.0
Вероятность захвата за точку риска, %	15 – 40**	25 – 50	5 – 20	< 2

Проведенный количественный сравнительный анализ демонстрирует преимущества предложенных алгоритмов относительно существующих аналогичных решений.

Литература

1. Siciliano, B., Khatib, O. (Eds.). Springer Handbook of Robotics. Springer, 2016.
2. Chaumette, F., Hutchinson, S. Visual servo control. I. Basic approaches // IEEE Robotics & Automation Magazine. 2006.
3. Redmon, J., et al. You Only Look Once: Unified, Real-Time Object Detection // CVPR. 2016.
4. Шевнина Ю.С., Гагарина Л.Г., Конюхов Е.В. Управление параметрами исполнительного оборудования при автоматизации производства // Известия высших учебных заведений. Электроника. 2024. Т. 29. № 3. С. 427-431.
5. Адаптивное управление совмещением и позиционированием элементов при автоматизации сборочных операций / Ю.С.. Шевнина [и др.] // Известия высших учебных заведений. Электроника. 2025. Т. 30. № 3. С. 303–312.

Y.S. Shevnina, E.I. Mirahmedow

National Research University of Electronic Technology (MIET)

DEVELOPMENT OF ALGORITHMS FOR CONTROLLING THE POSITION OF ROBOT MANIPULATOR ELEMENTS

Keywords: *robot manipulator, adaptive control, instance segmentation, visual servoing, computer vision, microelectronics, precision grasping.*

Abstract. *The article addresses the task of improving the accuracy and adaptability of a robot manipulator when working with microelectronic components. An analysis of the shortcomings of traditional control methods based on rigid trajectory programming and object detection via bounding boxes is presented. As a solution, a combined approach is proposed, including an instance segmentation algorithm for the precise determination of an object's position and orientation and an adaptive positioning algorithm with a proportional controller in a continuous vision-based feedback loop. The control task is formalized as a partial estimation of the object's 6D pose, and a system architecture is described that lays the foundation for scaling to three-dimensional manipulation tasks. The results of a comparative analysis demonstrating the advantages of the proposed method in positioning accuracy and robustness to object displacement are presented.*

Ю.С. Шевнина, М.Ю. Семенов, В.Д. Щербинин, А.А. Груздева
ООО «НМ-Tex»

УДК 517.968.23

ПОДХОД К ПРОГНОЗИРОВАНИЮ ОТКАЗОВ И ОПРЕДЕЛЕНИЮ ИХ ПРИЧИН В ПРОИЗВОДСТВЕННЫХ ПРОЦЕССАХ

Ключевые слова: *предиктивная аналитика, нейросетевые модели, ансамблирование, анализ причин брака, взвешенная линейная регрессия, производственные процессы.*

В работе представлен подход к прогнозированию отказов производственных процессов и анализу причин их возникновения. Рассмотрены нейросетевые модели различных типов (многослойный перцептрон, остаточная сеть, трансформер), объединённые в ансамбль для повышения точности прогноза. Для интерпретации результатов прогнозирования используется взвешенная линейная регрессию с экспоненциальным ядром и стандартизацию параметров.

Описаны математические модели, процедура обучения и сериализации компонентов, а также сценарии интеграции в производственную среду. Приведён пример настройки гиперпараметров модуля анализа причин и метрики качества.

1. Введение. Современные производственные процессы характеризуются высокой сложностью и большим количеством контролируемых параметров, что делает актуальной задачу автоматизированного прогнозирования отказов и анализа их причин. Традиционные методы статистического контроля часто не справляются с нелинейными зависимостями и многомерностью данных. В последние годы активно развиваются подходы на основе машинного обучения, в частности нейронных сетей, позволяющие строить точные прогнозы состояния производственных процессов [1]. Однако даже самый точный прогноз остаётся неполным без объяснения факторов, приведших к браку. Поэтому актуальна разработка интерпретируемых методов пост-анализа, которые могут дать технологу понятные рекомендации.

Целью данной работы является описание подхода к прогнозированию отказов производства, архитектуры и математического аппарата системы предиктивной аналитики. Программная реализация подхода ориентирована на интеграцию в промышленную среду с использованием реляционной СУБД и обеспечивает как онлайн-мониторинг, так и пост-обработку сменных данных.

2. Общее описание подхода. Рассматриваемый подход к прогнозированию отказов производственных процессов основан на принципах системного анализа и включает следующие основные этапы:

- сбор, предобработку и хранение производственных данных;
- прогнозирование параметров производственных процессов;
- анализ причин отказов производства;
- визуализацию полученных результатов.

Сбор данных осуществляется с датчиков микроклимата, контрольно-измерительных устройств (КЗИ), а также из корпоративной базы данных. После предобработки данные используются для обучения и инференса предиктивных моделей производственных процессов. Этап прогнозирования включает операции по определению вероятности возникновения отказа производства на основе текущих значений полученных параметров. При обнаружении аномалий осуществляется анализ причин отказов производства для определения вклада каждого параметра в увеличение риска. Результаты сохраняются в базе данных с возможной визуализацией и использованием для принятия решений.

3. Прогнозирование параметров производственных процессов основано на использовании ансамбля из трёх нейросетевых архитектур,

что позволяет повысить устойчивость и точность прогноза за счёт усреднения ошибок различных моделей [2].

Базовая линейная предиктивная модель представляет собой классический многослойный перцептрон с тремя скрытыми слоями (256, 128, 64 нейрона), после каждого из которых применяются пакетная нормализация результатов и рекуррентная конъюнкция с коэффициентом 0.3 для регуляризации модели.

Вторая модель в ансамбле основана на модели с остаточными связями и представляет собой глубокую сеть, состоящую из шести остаточных блоков. Каждый блок включает два полносвязных слоя с нормализацией и активацией, а также слой-соединение, суммирующий вход блока с его выходом. Такая архитектура решает проблему затухающих градиентов и позволяет строить модели большой глубины. Количество нейронов в блоках последовательно уменьшается: 512, 512, 256, 256, 128, 128. Для всех слоев применяется L2-регуляризация с коэффициентом $1e-4$.

Последняя модель представляет собой адаптацию архитектуры Transformer для табличных данных. Входной вектор проецируется в пространство размерности 512, преобразуется в трёхмерный тензор и подаётся на четыре последовательных блока трансформера. Каждый блок содержит многоголовое внимание (8 голов), остаточные связи и полносвязную сеть. После трансформеров данные уплотняются и проходят через два полносвязных слоя (256 и 128 нейронов) перед выходным слоем. Механизм внимания позволяет модели выявлять сложные взаимосвязи между признаками.

Ансамбль объединяет предсказания второй и третьей модели путём усреднения их выходов:

$$\text{ensemble} = \frac{P_{res} + P_{trans}}{2}. \quad (1)$$

Предложенная композиция моделей для прогнозирования отказов производственных процессов снижает дисперсию итогового прогноза и повышает обобщающую способность. Модели обучаются независимо на одних и тех же данных, но в силу различий архитектур они по-разному аппроксимируют целевую функцию, что делает их ошибки некоррелированными.

4. Анализ причин отказов производства. Определение причин отказов производственных процессов основано на локальной взвешенной линейной регрессии (Weighted Least Squares, WLS) с экспоненциальным ядром и предварительной стандартизацией параметров [3].

Для детекции случаев отказа производства используется алгоритм скользящего окна (размер $\text{lookback} = 40$). Для каждого момента времени t вычисляется минимальная вероятность брака в окне

$p_{\min} = \min(p[t - 40:t])$ и рост риска отказа производства $\text{rise} = p[t] - p_{\min}$.

Событие фиксируется, если выполняется хотя бы одно из условий:

- 1) резкий рост риска отказа производства: $\text{rise} \geq 0.12$;
- 2) превышение абсолютного порога вероятности отказа: $p[t] \geq 0.45$.

Для предотвращения дублирования вводится интервал охлаждения $\text{cool_down} = 60$ точек.

Локальная взвешенная регрессия. Для каждого обнаруженного события отказа строится локальная линейная модель, аппроксимирующая зависимость вероятности отказа производства от рассматриваемых параметров:

$$p(t) \approx \sum_{j=1}^d \beta_j x_j(t) + b, \quad (2)$$

где x_j – стандартизованные значения параметров, β_j – коэффициенты влияния, b – свободный член. Веса наблюдений задаются экспоненциальным ядром:

$$w_i = \exp\left(-\frac{|t_i - t_{\text{defect}}|}{\tau}\right), \tau = 20, \quad (3)$$

что придаёт больший вес точкам, близким к моменту возникновения отказа производства. Решение находится методом взвешенных наименьших квадратов с регуляризацией Тихонова с коэффициентом $\lambda = 10^{-6}$ для обеспечения численной устойчивости:

$$\beta = (X^T W X + \lambda I)^{-1} X^T W y. \quad (4)$$

Здесь $W = \text{diag}(w_i)$, X – матрица стандартизованных признаков.

Вклад каждого параметра в изменение вероятности отказов производственных процессов определяется как произведение его коэффициента β_j на отклонение от референсного состояния:

$$\text{contribution}_j = \beta_j \cdot (x_j(t_{\text{defect}}) - x_j(t_{\text{ref}})), \quad (5)$$

где t_{ref} – среднее значение параметра в первой половине окна анализа (нормальный режим). Для получения относительной важности в процентах выполняется нормировка:

$$\text{contribution_pct}_j = \frac{|\text{contribution}_j|}{\sum_{k=1}^d |\text{contribution}_k|} \cdot 100\%. \quad (6)$$

В базу данных сохраняются только параметры с вкладом не менее 5% (порог настраивается).

5. Пример работы и настройка гиперпараметров. На основе рассмотренного подхода к прогнозированию отказов производственных процессов осуществлена программная реализация в виде информационной системы, состоящей из модулей сбора данных, мониторинга и прогнозирования параметров производства. Модуль анализа причин отказов протестирован на исторических данных производства электронных компонентов. Основная нейросеть

(ансамбль) обучалась на выборке из 50 000 измерений 11 параметров (8 технологических, 3 климатических). При инференсе на тестовой выборке детектировано 24 события отказа производства. Для каждого события построена локальная регрессия и определены наиболее значимые параметры.

Калибровка пороговых значений проводилась по метрикам precision и recall с использованием экспертных оценок реальных причин отказов производства. Наилучшее сочетание (precision = 0.87, recall = 0.82) достигнуто при rising_delta = 0.10, threshold = 0.42 и min_contribution_pct = 5.0. Топ-1 точность совпадения главного параметра с экспертным мнением составила 0.75.

6. Заключение. В работе представлен подход к прогнозированию отказов производственных процессов, рассмотрена архитектура системы предиктивной аналитики, сочетающая нейросетевой ансамбль для прогнозирования отказов и интерпретируемый модуль анализа причин брака на основе взвешенной линейной регрессии. Экспериментальная проверка подтвердила эффективность предложенного подхода: ансамблирование повышает точность прогноза, а метод WLS даёт наглядное объяснение причин отказов производства, что позволяет операторам и лицам, принимающим решение, целенаправленно корректировать производственные параметры. Дальнейшие исследования будут направлены на автоматический подбор гиперпараметров модуля анализа причин отказов и расширение библиотеки интерпретирующих моделей.

Литература

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. М.: ДМК Пресс, 2018. 652 с.
2. Dietterich T.G. Ensemble methods in machine learning // International workshop on multiple classifier systems. Springer, 2000. P. 1–15.
3. Шевнина Ю.С. Моделирование состояния технологического оборудования в составе информационно-управляющей системы // Приборы и системы. Управление, контроль, диагностика. 2024. № 1. С. 29-35.
4. Шевнина Ю.С. Автоматический поиск решения в информационно-управляющей системе для контроля состояния производственной среды // Интеллектуальные системы в производстве. 2024. Т. 22. № 2. С. 19-25.

Yu.S. Shevnina, M.Yu. Semenov, V.D. Shcherbinin, A.A. Gruzdeva
LLC «NM-Tech»

AN APPROACH TO PREDICTING FAILURES AND DETERMINING THEIR CAUSES IN PRODUCTION PROCESSES

Keywords: *predictive analytics, neural network models, ensembling, defect cause analysis, weighted linear regression, manufacturing processes.*

Abstract. *The paper presents the architecture of a predictive analytics software system for forecasting equipment failures and analyzing the causes of defects in manufacturing processes. The core of the system is an ensemble of neural network models of different types (multilayer perceptron, residual network, transformer) combined to improve prediction accuracy. To interpret the predictions, a defect cause analysis module based on locally weighted linear regression with an exponential kernel and parameter standardization is developed. Mathematical models, training and serialization procedures, as well as integration scenarios into the industrial environment are described. An example of hyperparameter tuning for the cause analysis module and quality metrics is provided.*

Ю.С. Шевнина, С.Ф. Царапкин
Национальный исследовательский университет «МИЭТ»

УДК 004.9

ОЦЕНКА ПАРАМЕТРИЧЕСКОЙ ЧУВСТВИТЕЛЬНОСТИ МЭМС-ДАТЧИКОВ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Ключевые слова: *параметрическая чувствительность, МЭМС-датчики, мониторинг микроклимата, системы поддержки принятия решений, информационно-измерительные системы.*

В статье предложен и математически обоснован подход к оценке параметрической чувствительности ИИС, используемых для мониторинга микроклимата в ЧПП микроэлектроники. Показано, что для малых отклонений параметров датчиков целесообразно использовать не полные временные функции, а характеристические параметры. На примере МЭМС-датчика вибрации, описываемого моделью осциллятора второго порядка, получены аналитические зависимости граничной частоты от коэффициента затухания.

1. Введение. В современном микроэлектронном производстве соблюдение строгих параметров микроклимата в чистых производственных помещениях (ЧПП) является критическим фактором, влияющим на процент выхода годных изделий и надежность продукции. Системы поддержки принятия решений (СППР), управляющие инженерной инфраструктурой ЧПП, опираются на данные распределенной сети информационно-измерительных систем (ИИС).

Однако точность этих данных зависит от стабильности физических параметров первичных преобразователей (датчиков). В условиях

реальной эксплуатации параметры датчиков подвержены дрейфу вследствие старения, температурных влияний или технологических допусков. Актуальность исследования обусловлена необходимостью разработки математического аппарата, позволяющего СППР в реальном времени оценивать достоверность измерений и прогнозировать поведение измерительных каналов (в частности, вибрационных МЭМС-датчиков) при малых отклонениях их конструктивных параметров, что позволит предотвратить ложные срабатывания систем аварийной защиты и повысить качество технологического процесса [1].

Проблема заключается в необходимости перехода от сложных функциональных зависимостей к набору характеристических параметров, позволяющих с достаточной точностью оценивать параметрическую чувствительность системы приближенными методами.

Поведение ИИС зависит от ряда параметров k_i , которые относятся как к статическим и динамическим характеристическим функциям и значениям, так и к информационно-теоретическим параметрам. Данные параметры могут быть представлены параметрическим семейством $\bar{k}_i$. Во многих случаях отклонение параметра Δk_i от расчетного значения k_{i0} сравнительно невелико $\Delta k_i / k_{i0} \ll 1$.

2. Расчет параметрической чувствительности. Пусть измерительное устройство характеризуется передаточной функцией $G(p)$, весовой функцией $g(t)$ или функцией $h(t)$. Функции $G(p)$, $g(t)$ и $h(t)$ зависят от параметров k_i системы, в частности затухания, массы, индуктивности, емкости, сопротивления. Таким образом, получаем $G(p; k_i)$, $g(t; k_i)$, $h(t; k_i)$ [2].

Известно, что для учета изменений параметра k_i по отношению к значению k_{i0} при условии $\frac{\Delta k_i}{k_{i0}} = \frac{k_i - k_{i0}}{k_{i0}} \ll 1$ используются приближенные рассуждения.

С учетом вышеизложенного для переходной функции при параметрическом семействе $\bar{k}_i$ получим

$$h(t; \bar{k}_i) = h(t; \bar{k}_{i0}) + \sum_{i=1}^m \left(\frac{\partial h(t; \bar{k}_i)}{\partial k_i} \right)_{k_i=k_{i0}} \Delta k_i, \quad (1)$$

при этом коэффициент параметрического воздействия $\nu = \left(\frac{\partial h(t; \bar{k}_i)}{\partial k_i} \right)_{k_i=k_{i0}}$ также является временной функцией, получающейся в результате решения дифференциального уравнения. Аналогично определяются зависимости в частотном диапазоне. Для этого к уравнению (1) следует применить преобразование Лапласа.

Для описания временной характеристики вместо необходимых для точных исследований характеристических функций $G(p)$, $g(t)$ или $h(t)$ предлагается использовать характеристические параметры. При этом из длительного переходного процесса определяется время установления режима t_E как характеристический параметр во временном представлении, в то время как в частотном представлении используется граничная частота f_g . Предлагаемый подход позволяет в случае параметрической чувствительности указывать гораздо более наглядные соотношения между зависимыми компонентами ИИС.

Граничная частота f_g измерительного устройства зависит от отдельных параметров k_i параметрического семейства $\bar{k}_i$

$$f_g(\bar{k}_i) = f_g(\bar{k}_{i0}) + \sum_{i=1}^m \left(\partial f_g(\bar{k}_i) / \partial k_i \right)_{k_i=k_{i0}} \Delta k_i \quad (2)$$

с коэффициентом параметрического воздействия для предельной частоты $\nu_{fgi} = \left(\partial f_g(\bar{k}_i) / \partial k_i \right)_{k_i=k_{i0}}$.

Для времени установления режима аналогичное уравнение:

$$t_E(\bar{k}_i) = t_E(\bar{k}_{i0}) + \sum_{i=1}^m \left(\partial t_E(\bar{k}_i) / \partial k_i \right)_{k_i=k_{i0}} \Delta k_i \quad (3)$$

с коэффициентом параметрического воздействия для времени установления режима $\nu_{tEi} = \left(\partial t_E(\bar{k}_i) / \partial k_i \right)_{k_i=k_{i0}}$.

Для преобразования данных зависимостей используется теорема квантования по уравнению $t_E = 1/2 f_g$.

$$f_g(\bar{k}_i) = \frac{1}{2t_E(\bar{k}_{i0})} + \frac{1}{1 + \frac{1}{t_E(\bar{k}_{i0})} \sum_{i=1}^m \nu_{tEi} \Delta k_i} = \frac{1}{2t_E(\bar{k}_{i0})} - \frac{1}{2t_E^2(\bar{k}_{i0})} \sum_{i=1}^m \nu_{tEi} \Delta k_i, \quad (4)$$

следовательно, искомая зависимость имеет вид $\nu_{fgi} = -\left(\nu_{tEi} / 2t_E^2(\bar{k}_{i0}) \right)$, $\nu_{tEi} = -\left(\nu_{fgi} / 2f_g^2(\bar{k}_{i0}) \right)$.

Рассмотрим подробнее применение полученных зависимостей.

3. Примеры оценки параметрической чувствительности в ИИС.

Для мониторинга вибрации в ИИС микроклимата ЧПП, как правило, используются МЭМС-датчики. Для описания их работы используется модель механического осциллятора второго порядка, характеристиками которой являются «пружина–масса–затухание» и опорная точка. Имеем дифференциальное уравнение $m\ddot{x} + k\dot{x} + cx = F$, откуда при частоте собственных колебаний ω_0 незатухающей системы $\omega_0 = \sqrt{c/m}$ и

нормированном затухании $D = \delta/\omega_0 = k/2m\omega_0$ получим передаточную функцию $G(p) = \frac{1/c}{1 + \frac{2D}{\omega_0}p + \frac{1}{\omega_0^2}p^2}$. Очевидно, что функция $G(p)$

зависит от двух параметров $k_1 = c$, $k_2 = D$.

Для определения влияния параметра $k_2 = D$ вычислим граничную частоту в зависимости от D . Используя амплитудную характеристику

$$|G(j\lambda)| = \frac{1/c}{\sqrt{(1-\lambda^2)^2 + 4D^2\lambda^2}} \text{ и введя обозначение } \lambda = \omega/\omega_0, \text{ получим}$$

$$(1-\lambda_g^2)^2 + 4D^2\lambda_g^2 = 2 \text{ или } 1/2.$$

Для отнесенной к собственной частоте предельной частоты $\lambda_g = \omega_g/\omega_0 = f_g/f_0$ при затухании на -3дБ получим следующее выражение

$$\frac{f_g}{f_0} = \lambda_g = \sqrt{1-2D^2 \pm \sqrt{2}\sqrt{1-2D^2(1-D^2)}}, \quad (5)$$

а при затухании на +3дБ соответственно

$$\frac{f_g}{f_0} = \lambda_g = \sqrt{1-2D^2 \pm \sqrt{4D^2(D^2-1) + (1/2)}}. \quad (6)$$

Данные уравнения позволяют установить зависимости для времени установления режима t_E в переходных процессах. Кроме того, уравнения показывают, что для одного и того же значения затухания параметрическая чувствительность будет иметь наименьшее значение и будет равна 0, поскольку

$$\frac{\nu_{fgD}}{f_0} = \frac{\partial f_g/f_0}{\partial D}. \quad (7)$$

Путем вычисления коэффициента параметрического воздействия в соответствии с уравнением (7) получим для относительной граничной частоты по (5) и (6), нормированной по f_0 :

$$\frac{\nu_{fgD}}{f_0} = \frac{-2D \pm \frac{4D^3 - 2D}{\sqrt{2}\sqrt{1-2D^2(1-D^2)}}}{\sqrt{1-2D^2 \pm \sqrt{2}\sqrt{1-2D^2(1-D^2)}}}, \quad (8)$$

$$\frac{\nu_{fgD}}{f_0} = \frac{-2D \pm \frac{4D^3 - 2D}{\sqrt{4D^2(D^2-1) + (1/2)}}}{\sqrt{1-2D^2 \pm \sqrt{4D^2(D^2-1) + (1/2)}}}. \quad (9)$$

Полученные уравнения позволяют определить оптимальное значение затухания $D_{\text{опт}} = 0,6$. Физический смысл оптимального значения затухания заключается в том, что для $D < D_{\text{опт}}$ амплитудная характеристика обнаруживает резко выраженные резонансные свойства и, следовательно, граничная частота снова падает. На практике значение параметра D , как правило, определяется свойствами демпфирующих элементов.

4. Заключение. Таким образом, использование коэффициентов параметрического воздействия позволяет количественно оценить вклад каждого элемента ИИС в итоговую погрешность измерения. Это дает возможность СППР компенсировать систематические ошибки, а также формировать рекомендации по калибровке и замене оборудования до наступления критических сбоев в мониторинге микроклимата.

Литература

1. Шевнина Ю.С. Моделирование состояния технологического оборудования в составе информационно-управляющей системы // Приборы и системы. Управление, контроль, диагностика. 2024. №1. С. 29-35.
2. Оценка устойчивости информационно-управляющей системы на основе параметров предиктивной модели переходного процесса / Ю.С.. Шевнина [и др.] // Приборы и системы. Управление, контроль, диагностика. 2025. № 6. С. 21-32.

Yu.S. Shevnina, S.F. Tsarapkin

National Research University of Electronic Technology (MIET)

EVALUATION OF PARAMETRIC SENSITIVITY OF MEMS SENSORS IN DECISION SUPPORT SYSTEMS

Keywords: *parametric sensitivity, MEMS sensors, microclimate monitoring, decision support systems, information and measuring systems.*

Abstract. *This article proposes and mathematically substantiates an approach to assessing the parametric sensitivity of intelligent measurement systems (IMS) used for microclimate monitoring in microelectronics. It is shown that for small deviations in sensor parameters, it is advisable to use characteristic parameters rather than full time functions. Using a MEMS vibration sensor described by a second-order oscillator model as an example, analytical dependences of the cutoff frequency on the attenuation coefficient are obtained.*

ОБ ОДНОМ СПОСОБЕ ВЫЧИСЛЕНИЯ ДЕТЕРМИНАНТА МНОГОМЕРНЫХ МАТРИЦ

Ключевые слова: *многомерные матрицы, детерминант, трансверсаль, разложение по минорам, разделяй и властвуй, параллельные вычисления, OpenMP.*

В статье рассматриваются особенности вычисления детерминанта p -мерной кубической матрицы порядка n методом Соколова. Показано, что полный перебор трансверсалей имеет факториально-экспоненциальную сложность. Построены и реализованы разложения типа «разделяй и властвуй», сводящие задачу к сумме произведений детерминантов меньших порядков и формирующие набор независимых подзадач, пригодных для OpenMP. Приведены оценки числа подзадач и времени; эксперименты на случайных целочисленных тензорах показывают ускорение на несколько порядков относительно полного перебора.

Введение. В настоящее время одним из основных направлений развития линейной алгебры является обобщение классических определений и методов исследования на многомерные случаи [3]. Достаточно часто для этих целей возникает необходимость вычисления детерминанта для p -мерных кубических матриц порядка n .

Наиболее популярным методом вычисления определителя для многомерных матриц является метод Н.П. Соколова [1], предложенный в середине прошлого века. Данный метод обобщает классическое определение детерминанта на многомерный случай посредством использования трансверсалей и многомерных перестановок. Однако из-за высокой вычислительной сложности прикладные особенности метода до сих пор являются недостаточно исследованными.

1. Постановка задачи. Рассмотрим p -мерную кубическую матрицу (тензор) $A = (a_{i_1 i_2 \dots i_p})$, $i_\alpha \in \{1, 2, \dots, n\}$. Простым сечением ориентации α назовём $(p-1)$ -мерную матрицу, получаемую фиксацией индекса $i_\alpha = c$ и пробеганием остальных индексов по $\{1, \dots, n\}$.

Пусть выбрано несколько элементов $a_{i_1 \dots i_p}$. Такой набор называется *трансверсальным*, если для любой ориентации $\alpha \in \{1, \dots, p\}$ среди выбранных элементов все значения i_α попарно различны (то есть никакие два элемента не попадают в одно и то же простое сечение этой

ориентации). Трансверсальный набор из n элементов называется *трансверсалью*.

Эквивалентное задание трансверсали удобно фиксировать по первой оси: положим $i_1 = i$ ($i = 1, \dots, n$) и зададим остальные индексы перестановками $\sigma_2, \dots, \sigma_p \in S_n$:

$$T(\sigma_2, \dots, \sigma_p) = \{(i, \sigma_2(i), \dots, \sigma_p(i)) \mid i = 1, \dots, n\}. \quad (1)$$

Тогда число трансверсалей равно $(n!)^{p-1}$, так как оно совпадает с числом наборов перестановок $(\sigma_2, \dots, \sigma_p)$.

Выберем множество альтернативных осей $\mathcal{A} \subseteq \{2, \dots, p\}$ мощности m (чётное). Для каждой трансверсали знак её вклада равен произведению знаков перестановок по альтернативным осям, а сам вклад – произведению n элементов матрицы, взятых по этой трансверсали. В принятом выше представлении это даёт формулу

$$\det(A) = \sum_{\sigma_2 \in S_n} \dots \sum_{\sigma_p \in S_n} \left(\prod_{\alpha \in \mathcal{A}} \operatorname{sgn}(\sigma_\alpha) \right) \prod_{i=1}^n a_{i, \sigma_2(i), \dots, \sigma_p(i)}. \quad (2)$$

Замечание. Далее будем рассматривать только случай чётного числа альтернативных осей, как в исходном определении. При фиксации $i_1 = i$ ось 1 соответствует тождественной перестановке и не изменяет знак слагаемого.

2. Решение задачи методом «разделяй и властвуй».

Рассмотрим два основных подхода, основанных на теоремах 4 и 6, установленных в [1].

Теорема 4 [1, глава 1]. *Выделим в детерминанте $|A|$ порядка n произвольные v сечений некоторой ориентации ($1 \leq v \leq n-1$). Тогда сумма произведений всех миноров порядка v , расположенных в этих сечениях, на их алгебраические дополнения равна $|A|$.*

Алгоритм разложения. Используя описанную выше теорему 4, можно представить задачу в следующем виде:

$$\det(A) = \sum \operatorname{sign} \cdot \det(A_v) \cdot \det(\bar{A}_v). \quad (3)$$

Таким образом мы получаем возможность свести задачу к некоторому набору подзадач меньшего размера. Важно отметить, что мы получаем некоторый набор независимых детерминантов либо независимых подзадач вида минор и дополнение, что может быть распределено на разные потоки.

Оценка времени. Число комбинаций подмножеств размера v равно $(C_n^v)^{p-1}$. Можно получить рекурсивную оценку времени работы:

$$T(n) = O((C_n^v)^{p-1} \cdot (T(v) + T(n-v))). \quad (4)$$

Далее алгоритм можно раскрывать рекурсивно, а можно подставить вариант с расчётом каждого детерминанта полным перебором трансверсалей. В обоих случаях можно получить, что оптимум достигается при разбиении $v = n/2$. Подставляя и преобразуя все слагаемые для рекурсии, получим:

$$T(n) = O\left(n \left(\frac{n!}{\left(\frac{n}{q}\right)! \left(\frac{n}{q}\right)! \left(\frac{n}{q}\right)! \dots} \right)^{p-1}\right), \quad (5)$$

что даёт значительное сокращение по времени.

Теорема 6 [1, глава 1]. Разобьём n сечений некоторой ориентации на q групп, выделив в v -ю группу n_v сечений так, что $\sum_{v=1}^q n_v = n$. Тогда алгебраическая сумма выражений вида

$$(-1)^{\sum_{k=1}^{q-1} s_k} M_1 M_2 \dots M_q$$

по всем трансверсальным совокупностям миноров $M_1, \dots, M_q$ (каждый M_v имеет порядок n_v) даёт исходный детерминант $|A|$.

При этом число таких выражений равно:

$$\left(\frac{n!}{n_1! n_2! \dots n_q!} \right)^{p-1}.$$

Алгоритм, основанный на теореме 6, аналогичен алгоритму, основанному на теореме 4.

Оценка времени. В частном случае равных блоков $n_v = k$, $q = n/k$ можно получить примерно следующую оценку

$$T(n; k) = O\left(n \cdot \frac{(n!)^{p-1}}{(k!)^{(q-1)(p-1)}}\right), \quad q = \frac{n}{k}, \quad (6)$$

а случай $q = 2$ (блоки v и $n - v$) сводится к теореме 4.

3. Описание вычислительных экспериментов и их результаты.

Для проведения экспериментов были написаны две вспомогательные утилиты [4]:

1. Генератор случайных матриц. По заданным значениям n и p он генерирует матрицу из n^p элементов из равномерного распределения. Матрицы сохраняются в файл. Для проведения эксперимента был сгенерирован набор матриц для каждой из указанных в таблицах 1–3 пары значений n и p .

2. Утилита для замера времени исполнения. Ее задача заключается в чтении матриц из файлов и запуске некоторого количества итераций для указанного алгоритма. Для каждой итерации замерялось время выполнения функции, вычисляющей детерминант. В экспериментах по очереди указывался каждый из алгоритмов и количество итераций ставилось равным 5.

Описание используемых алгоритмов

1. Полный перебор: оптимизированный перебор всех комбинаций перестановок (baseline). Генерируем все перестановки для $(1..n)$. Для каждой оси нужно выбрать одну из перестановок. Перебираем все получившиеся комбинации индексов. Такой подход легко распараллелить [2].

2. Разложение по теореме 4: разбиение на подзадачи и вычисление каждого слагаемого полным перебором.

3. Рекурсивное разложение по теореме 4: многократное разбиение на блоки с последующим вычислением подзадач полным перебором.

4. Разложение по теореме 6: разбиение на блоки (в экспериментах – размера 2 или 3), далее вычисление подзадач полным перебором.

Результаты эксперимента описаны в таблицах 1–3.

Таблица 1

Последовательное исполнение: медиана времени (мс)

Алгоритм	$p = 4$			$p = 6$	
	$n = 4$	$n = 5$	$n = 6$	$n = 4$	$n = 5$
Baseline	0.65	56.75	13 731.95	293.75	956 536
Th4	0.76	11.30	155.31	69.41	29 330
Th4-rec	0.91	12.02	154.51	69.33	29 274
Th6	0.96	11.30	154.43	71.28	29 167

Таблица 2

OpenMP, $p = 4$: медиана времени (мс) для 8 и 16 потоков

Алгоритм	$n = 4$		$n = 5$		$n = 6$	
	8	16	8	16	8	16
Baseline	0.25	0.21	12.47	6.90	1 648.06	1 188.50
Th4	0.28	0.28	2.32	1.28	19.09	16.75
Th4-rec	0.28	0.23	2.21	1.14	21.22	15.72
Th6	0.20	0.96	2.30	1.40	24.43	16.72

Таблица 3

OpenMP, $p = 6$: медиана времени (мс) для 8 и 16 потоков

Алгоритм	$n = 4$		$n = 5$		$n = 6$	
	8	16	8	16	8	16
Baseline	67.24	40.44	132 366	99 098	>50ч.	>50ч.
Th4	13.28	7.84	3 324	2 399	204 252	156 952
Th4-rec	14.12	7.19	3 336	2 494	211 052	155 130
Th6	15.39	8.69	3 296	2 423	205 489	153 291

Наблюдения

1. На малых значениях n базовый перебор работает достаточно быстро, но при росте количества элементов значительно растёт время

исполнения.

2. Для всех алгоритмов с разбиениями рост количества элементов сказывается не так значительно, как для baseline, но все равно приводит к быстрому увеличению длительности обработки (в 5–130 раз).

3. Алгоритмы, основанные на разбиениях, сокращают время выполнения на порядки, что доказывает теоретическую оценку уменьшения количества операций.

Вывод. Получены алгоритмы на основе теорем разложения, уменьшающие трудоёмкость вычисления детерминанта Соколова за счёт декомпозиции на подзадачи меньшего размера. Численные эксперименты подтверждают сокращение времени вычислений и эффективность параллельного исполнения.

Литература

1. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 175 с.
2. Калужнин Л.А., Суцанский В.И. Преобразования и перестановки. М.: Наука, 1979. 112 с.
3. Муха В.С. Анализ многомерных данных: монография. Минск: Технопринт, 2004. 368 с.
4. Сенчилов В.В., Григорьева Г.М., Ходченков В.Ю. О результатах пробного тестирования системы комплексного интеллектуального анализа медицинских данных // Системы компьютерной математики и их приложения. Смоленск: Изд-во СмолГУ, 2020. Вып. 21. С. 206-213.

V.V. Senchilov, A.A. Filatenkov
Smolensk State University

ABOUT ONE METHOD FOR CALCULATING THE DETERMINANT OF MULTIDIMENSIONAL MATRICES

Keywords: *multidimensional matrices, determinant, transversal, minor decomposition, divide&conquer, parallel computing, OpenMP.*

Abstract: *This article examines the computation of the determinant of a p -dimensional cubic matrix of order n using Sokolov's method. It is shown that exhaustive search of transversals has factorial-exponential complexity. Divide-and-conquer decompositions are constructed and implemented, reducing the problem to a sum of products of determinants of lower orders and forming a set of independent subproblems suitable for OpenMP. Estimates for the number of subproblems and the computation time are provided; experiments on random integer tensors show a speedup of several orders of magnitude relative to exhaustive search.*

АРХИТЕКТУРА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Ключевые слова: *система рекомендаций, ALS, FastAPI, коллаборативная фильтрация, implicit.*

В статье предлагается архитектура рекомендательной системы общего назначения, предназначенной для обработки больших наборов данных. Архитектура реализует метод коллаборативной фильтрации, использующий многопоточную процедуру машинного обучения. Обоснован выбор библиотеки, выполняющей параллельные матричные операции и механизм использования высокопроизводительного фреймворка FastAPI, обеспечивающего работу системы в реальном времени. Благодаря высокой скорости и асинхронности, FastAPI создает качественное взаимодействие веб-сервиса с клиентскими приложениями PWA и React Native. Приведено описание вычислительного эксперимента, в котором была реализована и подвергнута анализу система рекомендации кинофильмов на основе эмоциональной реакции пользователя. Экспериментальная оценка показала точность прогнозирования $RMSE = 0.87$. Система поддерживает стабильную работу до 200 одновременных пользователей с временем ответа менее 600 мс.

1. Введение. Современные платформы цифрового контента сталкиваются с проблемой информационной перегрузки [1]; например, количество доступных кинофильмов растёт экспоненциально, затрудняя пользователям поиск релевантного контента [2]. Рекомендательные системы решают эту проблему [3], персонализируя выдачу контента и повышая вовлечённость. Однако внедрение таких систем сопряжено с техническими вызовами: алгоритмы машинного обучения требуют значительных вычислительных ресурсов, тогда как веб-приложения ожидают ответа за сотни миллисекунд. Это противоречие между сложностью моделей и требованиями к производительности определяет актуальность разработки архитектур, обеспечивающих работу рекомендаций в реальном времени. Цель настоящей статьи – разработка архитектуры рекомендательной системы, обеспечивающей высокую производительность за счёт разделения процессов обучения модели и

генерации рекомендаций, с поддержкой клиентских приложений PWA и React Native [4].

2. Описание архитектуры системы рекомендаций. Архитектура построена по принципу разделения обучения и обслуживания (training-serving split) (рис. 1), что позволяет оптимизировать компоненты независимо. Система включает четыре основных компонента: хранилище данных, обучающую систему, хранилище модели и веб-сервис реального времени.

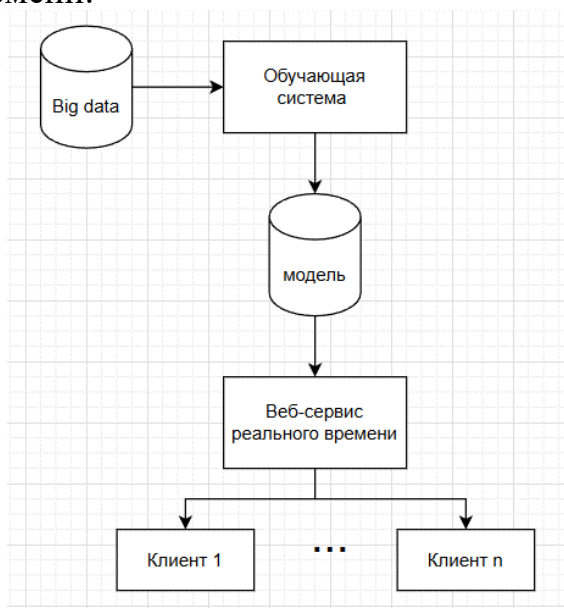


Рис. 1. Архитектура системы

Хранилище данных содержит исторические данные о взаимодействиях пользователей с фильмами и метаданные фильмов с эмоциональными тегами. Данные представляют разреженную матрицу «пользователь–фильм», где количество известных оценок составляет малую долю от всех возможных пар.

Обучающая система реализована в отдельном скрипте и выполняет загрузку и предобработку данных через библиотеку `pandas`, построение разреженной матрицы (рис. 2) `confidence` с использованием `scipy.sparse`, обучение модели ALS [5] через библиотеку `implicit` и сериализацию модели в файл `implicit_model.pkl`. Обучение выполняется офлайн, что исключает влияние на время отклика системы.

Хранилище модели представляет собой сериализованный `pickle`-файл, содержащий обученную модель ALS и отображения идентификаторов пользователей и фильмов. Это решение позволяет мгновенно загружать модель в память веб-сервиса без необходимости повторного обучения при каждом запросе.

Веб-сервис реализован на фреймворке FastAPI [6] и обеспечивает REST API с эндпоинтами (точками доступа). Сервис выполняет валидацию входных параметров, настройку CORS-политик для

безопасного взаимодействия с клиентскими приложениями, запуск ML-ядра через модуль subprocess без блокировки асинхронного цикла событий и преобразование результатов в формат JSON. Ключевой архитектурной особенностью является запуск ML-ядра как отдельного подпроцесса с указанием рабочей директории (cwd), что обеспечивает корректный поиск файлов данных независимо от места запуска API.

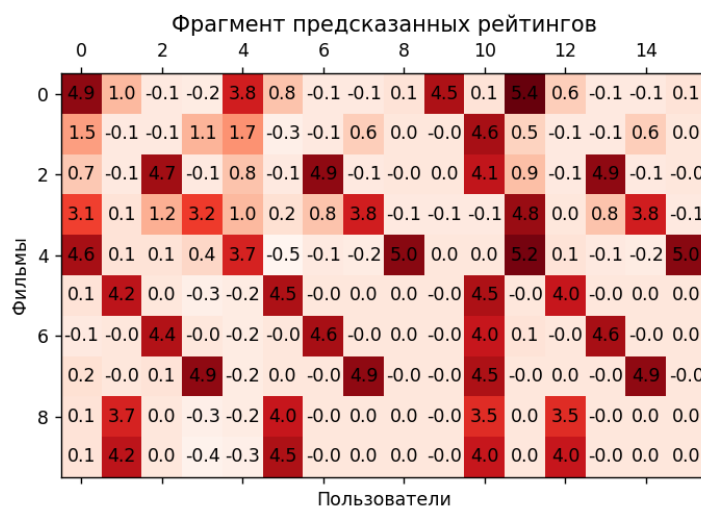


Рис. 2. Визуализация фрагмента матрицы предсказанных рейтингов

Процесс обработки запроса начинается с отправки клиентским приложением (PWA и React Native) HTTP GET-запроса на эндпоинт /recommendations с параметром настройки. FastAPI валидирует параметры и запускает скрипт обучающей системы. ML-ядро загружает модель, генерирует рекомендации алгоритмом ALS, применяет пост-фильтрацию по эмоциональному настроению. Результат записывается в файл, считывается API и возвращается клиенту в формате JSON. Такая схема изолирует тяжёлые вычисления от веб-сервера, обеспечивая стабильность при нагрузке.

3. Обоснование выбора программного средства. Выбор технологического стека связан с требованием высокой производительности и эффективной работы с данными в условиях ограниченных ресурсов. Для реализации серверной части выбран фреймворк FastAPI как один из самых быстрых Python-фреймворков, показывающий результаты, сопоставимые с NodeJS и Go. Нативная поддержка асинхронного программирования через async/await позволяет эффективно обрабатывать I/O-операции без блокировки потока выполнения. Интеграция с библиотекой Pydantic обеспечивает автоматическую проверку типов и валидацию входных данных. Встроенная генерация документации OpenAPI/Swagger UI упрощает интеграцию с фронтенд-приложениями. Встроенные middleware для

настройки CORS позволяют безопасно интегрировать API с PWA и React Native, размещёнными на других доменах или портах.

Для машинного обучения выбрана библиотека `implicit` с алгоритмом Alternating Least Squares. Библиотека специально разработана для работы с неявными предпочтениями, где отсутствие взаимодействия не означает отрицательного предпочтения. Использование структур данных `scipy.sparse` позволяет эффективно обрабатывать матрицы с заполненностью менее 1%. Реализация ALS использует библиотеки линейной алгебры OpenBLAS или Intel MKL, которые автоматически распараллеливают матричные операции на все доступные ядра процессора. Алгоритм ALS хорошо параллелится, устойчив к разреженности данных и сходится к глобальному минимуму функции потерь.

Для взаимодействия между веб-сервером и ML-ядром выбран подход с запуском отдельного процесса через модуль `subprocess`. По сравнению с альтернативами, `subprocess` требует минимальной инфраструктуры и упрощает развёртывание. Библиотеки `pandas` и `NumPy` выбраны для работы с данными благодаря векторизованным операциям и высокоуровневым структурам данных `DataFrame`. Модуль `pickle` выбран для сериализации модели благодаря простоте использования и быстрой загрузке в память.

4. Вычислительный эксперимент. Для оценки эффективности предложенной архитектуры был проведен эксперимент на специально подготовленном датасете, состоящем из двух взаимосвязанных файлов: `ratings.csv` (рейтинг пользователей) и `products.csv` (метаданные фильмов). Файл `products.csv` (рис. 3) включает информацию о 200 фильмах с полями: уникальный идентификатор, название фильма, жанровая классификация.

```
product_id,title,genres
1,"История игрушек (1995)","Adventure|Animation|Children|Comedy|Fantasy"
2,"Джуманджи (1995)","Adventure|Children|Fantasy"
3,"Старики-греки 2 (1995)","Comedy|Romance"
4,"Жди меня дома (1995)","Comedy|Drama|Romance"
5,"Невеста отца 2 (1995)","Comedy"
6,"Хитрость (1995)","Action|Crime|Thriller"
7,"Сабрина (1995)","Comedy|Romance"
```

Рис. 3. Данные файла `products.csv`

Файл `ratings.csv` (рис. 4) содержит 2200 записей пользовательских оценок и включает в себя следующие поля: идентификатор пользователя, идентификатор фильма, числовая оценка по пятибалльной шкале с шагом 0.5, временная метка в формате Unix epoch (охватывает период с января 2022 по январь 2026).

```

userId,productId,rating,timestamp
1,1,5.0,1642345200
1,2,4.5,1642431600
1,3,4.0,1642518000
1,4,5.0,1642604400

```

Рис. 4. Данные файла ratings.csv

Оценка проводилась по метрикам RMSE для точности предсказания, времени отклика сервера и пропускной способности при различной нагрузке.

Результаты показали точность $RMSE = 0.87$ на тестовой выборке (20% данных), что подтверждает эффективность алгоритма ALS на разреженных данных. Среднее время отклика составило 580 мс, максимальная стабильная нагрузка – 200 одновременных пользователей. Использование ресурсов при пиковой нагрузке: CPU 65-80% (4 ядра), RAM 1.2 ГБ. Структура времени отклика: загрузка модели 120 мс, генерация рекомендаций 280 мс, пост-обработка 80 мс, остальное 100 мс.

Тестирование клиентских приложений PWA и React Native подтвердило корректную работу CORS-политик и стабильное получение рекомендаций через единый REST API (рис.5). Время рендеринга на клиенте не превышало 100 мс.

№	Название фильма	Жанры
1	Последний богатырь (2017)	Приключения Комедия Фэнтези
2	Чебурашка (2023)	Приключения Детский Комедия Family
3	Солнцепёк (2023)	Комедия Драма
4	Паразиты (2019)	Комедия Драма Триллер
5	Джентльмены (2019)	Боевик Комедия Криминал
6	Шазам! (2019)	Боевик Комедия Фэнтези
7	Барби (2023)	Приключения Комедия Фэнтези
8	Стражи Галактики 2 (2017)	Боевик Приключения Комедия Фантастика
9	Один дома (1990)	Детский Комедия
10	Анти-герой (2021)	Боевик Приключения Комедия Фантастика

Рис. 5. Рекомендации, выданные системой

6. Заключение. Разработанная архитектура рекомендательной системы реального времени продемонстрировала эффективность разделения процессов обучения и обслуживания. Интеграция алгоритма ALS с постфильтрацией по эмоциональным критериям обеспечила

точность прогнозирования $RMSE = 0.87$. Веб-сервис на базе FastAPI с изоляцией вычислений через subprocess поддерживает до 200 одновременных пользователей со временем ответа менее 600 мс. Единый REST API обеспечивает корректную работу клиентских приложений на PWA и React Native. Предложенное решение подходит для средних веб-приложений благодаря простоте развёртывания и высокой производительности.

Литература

1. Shahrzadi L. et al. Causes, consequences, and strategies to deal with information overload: A scoping review // International Journal of Information Management Data Insights. 2024. Т. 4, №. 2. С. 100261.
2. Benson C.H., Okolo C.H., Oke O. The streaming-driven content evolution model: Impact of digital platforms on global film production // International Journal of Applied Research in Social Sciences. 2025. Т. 7, №. 8. С. 499-521.
3. Garapati R., Chakraborty M. Recommender systems in the digital age: a comprehensive review of methods, challenges, and applications // Knowledge and Information Systems. 2025. Т. 67, №. 8. С. 6367-6411.
4. React Native // URL: <https://reactnative.dev/>
5. Alternating Least Squares. URL: <https://arxiv.org/pdf/1509.00311>
6. FastAPI. URL: <https://fastapi.tiangolo.com/>

A.A. Nesterova, T.A. Samoilova
Smolensk State University

ARCHITECTURE OF A REAL-TIME RECOMMENDATION SYSTEM

Keywords: *recommendation system, ALS, FastAPI, collaborative filtering, implicit.*

Abstract. *The article proposes an architecture for a general-purpose recommender system designed to handle large datasets. The architecture combines a collaborative filtering method using a multi-threaded machine learning procedure. The choice of a library that performs parallel matrix operations and the use of the high-performance FastAPI framework, which ensures real-time system operation, is justified. Due to its high speed and asynchronous nature, FastAPI creates a high-quality interaction between the web service and PWA and React Native client applications. The paper describes a computational experiment in which a system for recommending movies based on user emotional response was implemented and analyzed. The experimental evaluation showed an RMSE prediction accuracy of 0.87. The system provides stable operation for up to 200 simultaneous users with a response time of less than 600 ms.*

СВЕДЕНИЯ ОБ АВТОРАХ

Абрамов Глеб Олегович – студент Смоленского государственного университета. E-mail: gabramov723@gmail.com.

Акимов Максим Александрович – магистрант физико-математического факультета Смоленского государственного университета. E-mail: m.akimovav@mail.ru.

Андреев Артем Ренатович – аспирант кафедры радиоэлектронных систем и комплексов Российского технологического университета МИРЭА. E-mail: artem09594@yandex.ru.

Аникин Иван Сергеевич – старший преподаватель Национального исследовательского университета «Московский институт электронной техники». E-mail: e-mail:anikin_is@mail.ru.

Антонов Кирилл Алексеевич – инженер (стажер) ООО «НМ-Тех». E-mail: kantonov@nm-tech.org.

Бирюкова Татьяна Константиновна – кандидат физико-математических наук, старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук, лауреат Премии Правительства Российской Федерации в области науки и техники. E-mail: yukonta@mail.ru.

Бондарь Александр Александрович – кандидат технических наук, доцент кафедры радиоэлектронных систем и комплексов Российского технологического университета МИРЭА. E-mail: bondar_a@mirea.ru.

Волков Александр Сергеевич – старший преподаватель института СПИНТех НИУ «МИЭТ». E-mail: alex.volkov9595@inbox.ru

Вологжанин Никита Андреевич – магистрант института СПИНТех, НИУ «МИЭТ». E-mail: vologzhanin.nik@mail.ru.

Гагарина Лариса Геннадьевна – доктор технических наук, профессор, директор института системной и программной инженерии и информационных технологий (СПИНТех) НИУ «МИЭТ». E-mail: gagar@bk.ru

Ганин Матвей Алексеевич – аспирант кафедры радиоэлектронных систем и комплексов МИРЭА – Российский технологический университет. E-mail: matthewganin@mail.ru.

Гершкович Максим Михайлович – старший научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук, лауреат Премии Правительства Российской Федерации в области науки и техники. E-mail: makmg@mail.ru.

Голубева Ирина Алексеевна – магистрант института СПИНТех НИУ «МИЭТ». E-mail: irinkagolubeva.com@gmail.com.

Гончаров Евгений Игоревич – аспирант кафедры прикладной математики и информатики Смоленского государственного университета. E-mail: drbenvey1996@mail.ru.

Городилов Алексей Владиславович – кандидат технических наук, институт СПИНТех НИУ «МИЭТ». E-mail: kaverina@mail.ru.

Груздева Анастасия Алексеевна – инженер (стажер) ООО «НМ-Тех». E-mail: gruzdeva@nm-tech.org.

Захаров Виктор Николаевич – доктор технических наук, доцент, ученый секретарь Федерального исследовательского центра «Информатика и управление» Российской академии наук. E-mail: vzakharov@frccsc.ru.

Игнатова Ирина Гургеновна – научный руководитель института СПИНТех НИУ «МИЭТ», доктор технических наук, профессор. E-mail: igi@micee.ru.

Карпенко Анатолий Павлович – доктор физико-математических наук, профессор, заведующий кафедрой САПР МГТУ им. Н.Э. Баумана. E-mail: apkarpenko@bmstu.ru.

Киселева Алина Константиновна – студент Смоленского государственного университета. E-mail: alinoker3000@yandex.ru.

Козлов Сергей Валерьевич – кандидат педагогических наук, доцент, заведующий кафедрой прикладной математики и информатики Смоленского государственного университета. E-mail: svkozlov1981@yandex.ru.

Кононова Александра Игоревна – доктор физико-математических наук, профессор института СПИНТех НИУ «МИЭТ». E-mail: illinc@bk.ru.

Коробко Сергей Сергеевич – магистрант института СПИНТех НИУ «МИЭТ». E-mail: sergeikorobko554@gmail.com.

Кристалинский Владимир Романович – кандидат физико-математических наук, доцент, доцент кафедры прикладной математики и информатики Смоленского государственного университета, научный сотрудник Военной академии войсковой противовоздушной обороны ВС РФ. E-mail: kristvr@rambler.ru.

Кугоев Иван Александрович – соискатель института МПСУ, НИУ «МИЭТ». E-mail: evgen_uis@mail.ru.

Кудинов Владислав Владимирович – аспирант института радиоэлектроники и информатики Российского технологического университета. E-mail: vladik_kudinov@mail.ru.

Лопырёв Вячеслав Алексеевич – аспирант института СПИНТех НИУ «МИЭТ». E-mail: slopyryov@gmail.com.

Масленников Мстислав Владимирович – ведущий математик факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова. E-mail: steve.maslennikov@yandex.ru.

Машенков Артем Сергеевич – аспирант физико-математического факультета Смоленского государственного университета. E-mail: asrtemmashenkoff20158@gmail.com.

Медведев Дмитрий Сергеевич – студент Смоленского государственного университета. E-mail: jjba746blend@gmail.com.

Мирасов Вадим Фаритович – начальник аналитического отдела ООО «НМ-Тех». E-mail: mirasov@nm-tech.org.

Мирахмедов Эльдар Интигамович – магистрант института СПИНТех НИУ «МИЭТ». E-mail: eld.mirahmedow2014@yandex.ru.

Миронов Николай Сергеевич – старший преподаватель института СПИНТех НИУ «МИЭТ». E-mail: mniklser@yandex.ru.

Мунерман Виктор Иосифович – доктор технических наук, профессор кафедры прикладной математики и информатики Смоленского государственного университета. E-mail: vimoona@gmail.com.

Мунерман Даниил Викторович – стажер-исследователь кафедры прикладной математики и информатики Смоленского государственного университета. E-mail: danvmoon@gmail.com.

Назаренко Артём Александрович – младший аналитик ООО «НМ-Тех». E-mail: nazarenko@nm-tech.org.

Нестерова Анастасия Александровна – студент физико-математического факультета Смоленского государственного университета. E-mail: nastuaa.04@yandex.ru.

Никаноров Всеволод Дмитриевич – магистрант института СПИНТех НИУ «МИЭТ». E-mail: nikanorov.vd@yandex.ru.

Попов Алексей Вячеславович – кандидат технических наук, старший преподаватель института СПИНТех НИУ «МИЭТ». E-mail: alex_std_ex@mail.ru.

Попов Данила Андреевич – студент ФГБОУ ВО «Смоленский государственный университет». E-mail: neleemails@gmail.com.

Портнов Евгений Михайлович – доктор технических наук, профессор института СПИНТех НИУ «МИЭТ». E-mail: evgen_uis@mail.ru.

Рахимкулов Захар Максимович – магистрант института СПИНТех НИУ «МИЭТ». E-mail: zakhar@fishandfly.ru.

Редченков Дмитрий Станиславович – студент физико-математического факультета Смоленского государственного университета. E-mail: dima67rud@yandex.ru.

Савельев Максим Алексеевич – аспирант института СПИНТех НИУ «МИЭТ». E-mail: maksimklin2009@mail.ru.

Самойлова Татьяна Аркадьевна – кандидат технических наук, доцент, доцент кафедры прикладной математики и информатики

Смоленского государственного университета. E-mail: tatsamoilova24@gmail.com.

Сапрыкин Олег Константинович – аспирант Российского технологического университета МИРЭА. E-mail: oleg.sap01@yandex.ru.

Семенов Михаил Юрьевич – кандидат технических наук, директор научно-технического центра ООО «НМ-Тех». E-mail: semenov@nm-tech.org.

Сенчилов Владислав Владимирович – кандидат физико-математических наук, доцент кафедры прикладной математики и информатики Смоленского государственного университета. E-mail: senchilov.vv@gmail.com.

Си Ту Тант Син – кандидат технических наук, докторант института СПИНТех НИУ «МИЭТ». E-mail: sithuthantsin86@gmail.com.

Симаков Валерий Александрович – аспирант физико-математического факультета Смоленского государственного университета. E-mail: 67Rus.67Rus@gmail.com.

Синицин Владимир Игоревич – доктор физико-математических наук, главный научный сотрудник, руководитель отделения Федерального исследовательского центра «Информатика и управление» Российской академии наук, лауреат Премии Правительства Российской Федерации в области науки и техники. E-mail: sinitsin_vi@mail.ru.

Слюсарь Мария Владимировна – старший преподаватель института СПИНТех НИУ «МИЭТ». E-mail: vslyusar@mail.ru.

Смирнов Владислав Сергеевич – студент факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: lilcube@inbox.ru.

Суханова Анна Геннадьевна – кандидат технических наук, доцент, доцент 14 кафедры (естественно-научных дисциплин) Военной академии войсковой противовоздушной обороны ВС РФ. E-mail: Ann-Sukhanova@yandex.ru.

Титов Юрий Павлович – кандидат технических наук, доцент, доцент кафедры 304 МАИ «Вычислительные машины, системы и сети». E-mail: kalengul@mail.ru.

Тихонова Ольга Вадимовна – доктор технических наук, старший научный сотрудник, профессор кафедры радиоэлектронных систем и комплексов Российского технологического университета МИРЭА. E-mail: o_tikhonova@inbox.ru.

Федотова Елена Леонидовна – кандидат педагогических наук, доцент, доцент института СПИНТех НИУ «МИЭТ». E-mail: fedotova-e2007@yandex.ru.

Федулова Светлана Александровна – заведующая лабораторией информатизации, старший преподаватель кафедры вычислительной

техники филиала Национального исследовательского университета «МЭИ» в г. Смоленске. E-mail: svfed67@mail.ru.

Филатенков Артур Андреевич – аспирант Смоленского государственного университета, E-mail: filatenkov.aa@phystech.edu

Царапкин Сергей Федорович – магистрант института СПИНТех НИУ «МИЭТ». E-mail: spintex@yandex.ru.

Чернышов Максим Карнельевич – кандидат физико-математических наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: mkch69@gmail.com.

Шевнина Юлия Сергеевна – доктор технических наук, доцент, профессор института СПИНТех НИУ «МИЭТ». E-mail: yusm@rambler.ru.

Щербинин Владислав Денисович – инженер-программист ООО «НМ-Тех». E-mail: shcherbinin@nm-tech.org.

Юдахин Юрий Юрьевич – аспирант института СПИНТех НИУ «МИЭТ». E-mail: yudakhin.yuri@gmail.com.

СОДЕРЖАНИЕ

РАЗДЕЛ 1. Системы компьютерной математики	3
<i>Андреев А.Р., Бондарь А.А., Тихонова О.В.</i> Разработка Raptor-кодера для реализации на ПЛИС	3
<i>Бирюкова Т.К., Гершкович М.М., Синицин В.И.</i> Вопросы идентификации при организации хранения и поиске информационных объектов с неточно известными мультимодальными атрибутами в территориально-распределенных информационно-телекоммуникационных системах	9
<i>Ганин М.А.</i> Дискретное преобразование Зака и операторная модель линейного нестационарного канала: интерпретация в домене задержка–доплер	18
<i>Кристалинский В.Р.</i> О реализации алгоритма приближенного вычисления интеграла типа Коши в системе Wolfram Mathematica	25
<i>Кудинов В.В.</i> Исследование влияния параметров циклического кода на эффективность итеративного декодирования в космических каналах связи	32
<i>Миронов Н.С.</i> Разработка модели глубокого обучения для выявления ошибок при сборке печатных плат	39
<i>Миронов Н.С., Юдахин Ю.Ю.</i> Разработка метода сегментации дефектов пайки печатных плат	48
<i>Сапрыкин О.К.</i> Сравнительный анализ CNN, LSTM и GRU для автоматической классификации модуляции сигналов	59
<i>Смирнов В.С., Чернышов М.К.</i> Развертывание корпоративного почтового сервера Mailcow: Dockerized в виртуальной сетевой лаборатории EVE-NG	65
<i>Суханова А.Г.</i> Исследование отношений между свойствами в системе Mathcad	72
<i>Тихонова О.В.</i> Различные типы моделей для разработки радиоэлектронных систем	77
<i>Федулова С.А.</i> Программные средства интеллектуальной поддержки принятия решений на основе нечеткого ситуационно-когнитивного подхода	84
<i>Юдахин Ю.Ю.</i> Методика оценки критичности дефектов корпусов интегральных микросхем в системах автоматизированного контроля	89
РАЗДЕЛ 2. Параллельные вычисления и многоядерные процессоры	95
<i>Абрамов Г.О., Медведев Д.С., Мунерман В.И., Попов Д.А.</i> Архитектура USB кластера для параллельного умножения матриц по алгоритму Кэннона	95
<i>Акимов М.А., Козлов С.В.</i> Система распределённых вычислений с динамической загрузкой задач	100
<i>Аникин И.С.</i> Разработка математической модели представления сетевого трафика	105

<i>Волков А.С., Портнов Е.М., Слюсарь М.В.</i> Метод интеллектуальной диагностики неисправностей цифровых подстанций	110
<i>Вологжанин Н.А., Гагарина Л.Г., Голубева И.А.</i> Динамическое моделирование литографии: анализ чувствительности паттернов к вариациям параметров с использованием Fourier Neural Operators (FNO)	120
<i>Гагарина Л.Г., Коробко С.С., Рахимкулов З.М.</i> Разработка модульной архитектуры программного обеспечения автоматизированного контроля	127
<i>Гагарина Л.Г., Савельев М.А.</i> Автоматизированная информационно-измерительная система для испытаний антенн в безэховой камере с применением методов искусственного интеллекта	131
<i>Гончаров Е.И.</i> Математические аспекты Web3.0	139
<i>Городилов А.В., Кононова А.И.</i> Кинематическая модель манипулятора	144
<i>Захаров В.Н.</i> Виртуальный компьютерный музей как источник знаний по вычислительной технике и информационным технологиям	149
<i>Игнатова И.Г., Слюсарь М.В.</i> Обнаружение и классификация неисправностей в электрических сетях на основе алгоритма Байеса–СДКП–самовнимания	156
<i>Карпенко А.П.</i> Новые методы диверсификации поиска в популяционных алгоритмах глобальной оптимизации. Обзор	166
<i>Киселева А.К., Мунерман В.И.</i> Алгоритмическая теория случайности в практическом анализе данных	184
<i>Кугоев И.А.</i> Разработка метода балансировки нагрузки в облачных системах на основе модифицированного алгоритма Falcon Optimization	189
<i>Лопырёв В.А., Гагарина Л.Г.</i> Разработка алгоритма параллельного сканирования сжатых колоночных данных с декомпрессией и декодированием	198
<i>Масленников М.В.</i> Сквозной (λ, μ) -анализ и структурное распараллеливание вычислений в архитектуре Mamba	206
<i>Машенков А.С., Мунерман В.И.</i> Оптимизация последовательности произведения многомерных матриц	213
<i>Мирасов В.Ф., Антонов К.А., Назаренко А.А.</i> Методика распознавания дефектов изделий микроэлектроники с использованием предобученных моделей	218
<i>Мунерман В.И., Мунерман Д.В.</i> Алгебраический подход к представлению структурированных данных	222
<i>Никаноров В.Д., Шевнина Ю.С., Федотова Е.Л.</i> Предиктивное управление нелинейной системой на основе оператора Купмана	227
<i>Попов А.В.</i> Моделирование процессов функционирования сетей связи в условиях масштабируемости	232
<i>Редченков Д.С., Самойлова Т.А.</i> Анализ моделей параллелизма в системах взаимодействия клиент-сервера	237

<i>Самойлова Т.А.</i> Метод создания веб-сервиса для классификации изображений в режиме реального времени	242
<i>Си Ту Тант Син.</i> Модель мультиэшелонной системы управления запасами и пополнения ресурсов	250
<i>Си Ту Тант Син.</i> Моделирование цифровых двойников для систем мультиэшелонного управления процессами ресурсного обеспечения	258
<i>Симаков В.А.</i> Применение соглашения Эйнштейна для эффективной реализации (λ, μ) -свернутого произведения многомерных матриц	268
<i>Титов Ю.П.</i> Гибридный подход к решению задач дискретной оптимизации: эффективное использование SIMD и SIMT архитектур в матричном методе муравьиных колоний	274
<i>Шевнина Ю.С., Мирахмедов Э.И.</i> Разработка алгоритмов управления положением элементов робота-манипулятора	286
<i>Шевнина Ю.С., Семенов М.Ю., Щербинин В.Д., Груздева А.А.</i> Подход к прогнозированию отказов и определению их причин в производственных процессах	291
<i>Шевнина Ю.С., Царапкин С.Ф.</i> Оценка параметрической чувствительности МЭМС-датчиков в системах поддержки принятия решений	296
<i>Сенчилов В.В., Филатенков А.А.</i> Об одном способе вычисления детерминанта многомерных матриц	301
<i>Нестерова А.А., Самойлова Т.А.</i> Архитектура рекомендательной системы реального времени	306
Сведения об авторах	312

Системы компьютерной математики и их приложения

Выпуск 27

Часть 1

Издательство Смоленского государственного
университета

Редакторы *О.В. Папко, И.В. Марусова, И.А. Казилина*

Подписано к печати 22.04.2026. Формат 60×84 1/16.

Бумага офсетная. Печать ризографическая.

Усл. п. л. 20,0. Уч.-изд. л. 20,0. Тираж 300 экз.

Заказ №

Отпечатано с готового оригинал-макета

ИП Суркова Н.Н.

214000 г. Смоленск, ул. Б. Советская, д. 12/1

Тел.: 8-951-697-22-23