ISSN 2220-3397

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Программная инженерия

ПрИн
1
2021
Том 12

Уважаемые авторы и читатели журнала «Программная инженерия»!

Прошел ещё один – 10-й год в жизненном цикле нашего журнала. По традиции, это время подведения итогов прошедшего года, формирования планов на год будущий.

По итогам прошлого года расширилась география авторского представительства журнала. Авторы представляют 17 регионов и более 30 вузов и научно-технических центров нашей страны и ближнего зарубежья. Увеличился в количественном и качественном отношении состав экспертов, принимавших участие в рецензировании представленных для публикации в нашем журнале статей. В среднем, на каждую статью приходится два-три рецензента. При сохранении высоких требований к публикациям по уровню представляемых авторами результатов, стилю их изложения, по соблюдению нормативов и рекомендаций, далеко не все статьи успешно проходят повторное двукратное рецензирование, а некоторые, к сожалению, не проходят вообще. Такой подход обеспечивает качество и практическую востребованность публикуемых в журнале материалов и настраивает «неудачников» на более ответственное отношение к представляемым в журнал статьям. Несмотря на уменьшение объёма журнала и числа публикуемых статей, его импакт-фактор в среднем удерживается на высоком уровне, соответствующем журналам аналогичного научно-технического направления.

В журнале за последний год вышли публикации, тематика которых соответствует перечню первоочередных Национальных проектов и критических технологий, которые обозначены в документах на государственном уровне. К их числу относятся «Цифровая экономика РФ», «Большие данные и искусственный интеллект», «Квантовые технологии», «Информационная безопасность» и др. В таких статьях представляются результаты поисковых исследований и практических работ, затрагивающих весь жизненный цикл программных систем. Эти факторы – дополнительное свидетельство того, что мы на правильном пути.

Проведенные в прошедшем году первые опросы на большой выборке ученых, занимающихся вопросами, по тематике близкими к инженерии программ, дают основания для утверждения о том, что мы неплохо учитываем интересы авторов статей и наших читателей.

В планах на будущий год – продолжить все перечисленные выше тенденции. В условиях сложностей, которые обусловлены пандемией COVID-19 и её последствиями, сделать это будет непросто. Ситуация с издательским делом и отношением к подписке на журналы научно-технического профиля пока также оптимизма не добавляет. Нужно искать новые резервы и подходы, в первую очередь, на основе цифровизации издательских процессов. Однако, как показывает практика прошедших лет, мы такие подходы находим и вместе с журналом растём и развиваемся. Это залог наших успехов в будущем году.

Поздравляю Вас с Новым 2021 годом! Желаю каждому из Вас здоровья, счастья в семье, спокойствия, уверенности в будущем и удачи.

Главный редактор
д-р физ.-мат. наук, проф. **В. А. Васенин**

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с сентября 2010 г. DOI 10.17587/issn.2220-3397 ISSN 2220-3397

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство о регистрации ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индекс по Объединенному каталогу "Пресса России" — **22765**) или непосредственно в редакции.
Тел.: (499) 269-53-97. Факс: (499) 269-55-10.
Http://novtex.ru/prin/rus E-mail: prin@novtex.ru
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 12
N 1
2021

Published since September 2010 **DOI 10.17587/issn.2220-3397** **ISSN 2220-3397**

CONTENTS

Information about the jornal is available online at:
http://novtex.ru/prin/eng e-mail:prin@novtex.ru

УДК 004.4 DOI:10.17587/prin.12.3-7

П. А. Трунин, инженер, **Л. С. Прохоренко,** мл. науч. сотр., pro.leonid@gmail.com,
Д. С. Мищенков, мл. науч. сотр., **Д. Д. Климов,** ст. науч. сотр.,
Московский государственный технологический университет "СТАНКИН"

Разработка протоколонезависимой модульной архитектуры для обработки данных от трехкомпонентного датчика измерения силы

Описаны предложенные авторами подходы к разработке протоколонезависимой модульной библиотеки для обработки данных от трехкомпонентного датчика измерения силы. Рассмотрены различные архитектурные решения для разработки библиотеки. В рамках эксперимента проведено количественное и качественное сравнение реализаций, приведен анализ результатов. Даны практические советы и рекомендации.

Ключевые слова: *архитектура программного обеспечения, разработка программного обеспечения, практики разработки, поддерживаемое программное обеспечение, встраиваемые системы*

Введение

С развитием микроэлектроники все бóльшую популярность получают такие компактные энергоэффективные устройства, как акселерометры, магнитометры, гироскопические сенсоры, тензодатчики. Появление новых подобных сенсоров позволяет создавать принципиально новые устройства.

На базе подобных компактных сенсоров разрабатывают различные датчики, такие как тахометры, датчики силы, момента, IMU-датчики (инерционные измерительные устройства, датчики, позволяющие определить приращение положения и ориентации). Такие датчики могут применяться в разных сферах науки и техники. Например, широкое распространение в робототехнике [1—4], в том числе медицинской [5—7], и механообработке [8—10] получили датчики силы. В промышленной робототехнике датчики силы применяют при разработке рабочих органов, в том числе захватных устройств, в медицинской робототехнике данные датчики используют для очувствления роботов, они позволяют проводить сложные, динамически контролируемые операции [11, 12]. В механообработке датчики силы могут применяться для базирования деталей или измерения сил резания [13]. При решении задач автоматизации сборочных операций датчики силы могут быть использованы для определения сил в захватном устройстве или при оценке сопротивления во время сборки и разборки [14]. IMU-сенсоры получили широкое распространение в мобильной робототехнике [15] и в разработке летательных аппаратов [16]. Компактные тензодатчики применяют в том числе при создании шестикомпонентных силомоментных датчиков [17], также получивших широкое применение в робототехнике.

Такие датчики используют во встраиваемых системах, которые зачастую имеют достаточно строгие ограничения по памяти. В связи с этим обстоятельством большую роль играет размер программной части, предназначенной для передачи и обработки данных, получаемых от датчика.

В настоящей работе рассматривается выбор архитектуры протоколонезависимой библиотеки на языке C++ для обработки данных, передаваемых датчиком силы, основанном на MLX90393 Triaxis Magnetic Node (Melexis) — трехкомпонентном сенсоре магнитного поля. MLX90393 может работать с использованием различных протоколов передачи данных (SPI или I2C).

Протокол передачи данных от датчика на базе MLX90393 зависит от аппаратного исполнения самого датчика: в зависимости от технологической потребности датчик может быть изготовлен как в SPI-исполнении, так и в I2C-исполнении. Данные, получаемые от датчика, требуют предварительной обработки для дальнейшей их интерпретации как информации о прилагаемом к датчику усилии. В связи с этим необходима библиотека для обработки данных, учитывающая в том числе и разные исполнения датчика.

Выбор архитектуры библиотеки важен, так как он влияет на весь последующий процесс разработки, в том числе на предоставляемый API (программный интерфейс приложения). Библиотека была разработана на языке C++ поскольку данный язык предоставляет удобные инструментальные механизмы для объектно-ориентированного описания данных, использования шаблонов и имеет широкое представление в индустрии при разработке как встраиваемых систем, так и кроссплатформенного программного обеспечения [18].

Цель данной статьи — расширение прикладного кругозора разработчиков программного обеспечения для встраиваемых систем. Рассмотренные в статье принципы могут быть применены при самостоятельной разработке встраиваемых систем.

Постановка задачи исследования

Для выбора архитектуры библиотеки необходимо выявить качественные и количественные критерии оценки. При разработке библиотеки, ориентированной на использование во встраиваемых системах, количественными критериями можно считать размер компилируемых файлов и затраты памяти. К качественным критериям можно отнести простоту повторного использования кода, удобство внесения изменений и легкость, с которой он может быть интегрирован в другие библиотеки.

Задача исследования состоит в оценке различных архитектурных подходов на основе предложенных критериев.

Методы исследования

Для проведения исследования были разработаны три реализации библиотеки на языке C++ для работы с датчиком на базе сенсора MLX90393, позволяющие выбрать протокол передачи данных. Для исследования были выбраны следующие реализации.

- Две независимые версии, каждая версия отвечает за свой протокол передачи данных, обработка и передача данных объединены в единые программные модули. При такой организации библиотеки не накладываются ограничения на API независимых версий.
- Два независимых класса, отвечающих за передачу данных, один класс, отвечающий за обработку данных, написанный с использованием шаблонов. Шаблоны C++ это мощная языковая конструкция, позволяющая повторно использовать код. При такой организации библиотеки независимые классы должны обладать идентичным API.
- Два класса, отвечающих за передачу данных, объединенных одним абстрактным классом, один класс, отвечающий за обработку данных, написанный с использованием вызова виртуальных функций. В этом случае аппаратно-программный интерфейс классов, отвечающих за обработку данных, диктуется общим абстрактным классом.

Для каждой реализации библиотеки были скомпилированы следующие три программы: для обработки данных от одного датчика по протоколу SPI; для обработки данных от одного датчика по протоколу I2C; для обработки данных от двух датчиков по протоколам I2C и SPI. Также была разработана программа, не выполняющая получение и обработку данных. Эта программа использована для анализа затрачиваемой на реализацию приложений памяти, не связанной с передачей и обработкой данных. Программы были скомпилированы для микроконтроллера ATmega32U4 с использованием компилятора avr-g++ версии 7.3.0.

На рис. 1 представлена блок-схема программы для обработки данных, получаемых от одного датчика.

Все версии программ, работающих с одним или двумя датчиками, имеют общую принципиальную часть. Выполнение программы начинается с инициализации датчика, после чего следует вечный цикл. Программа работает на микроконтроллере, операции записи в энергонезависимую память не проводятся, в связи с чем завершение программы осуществляется "на горячую", посредством отключения питания. В вечном цикле вызывается процедура проведения

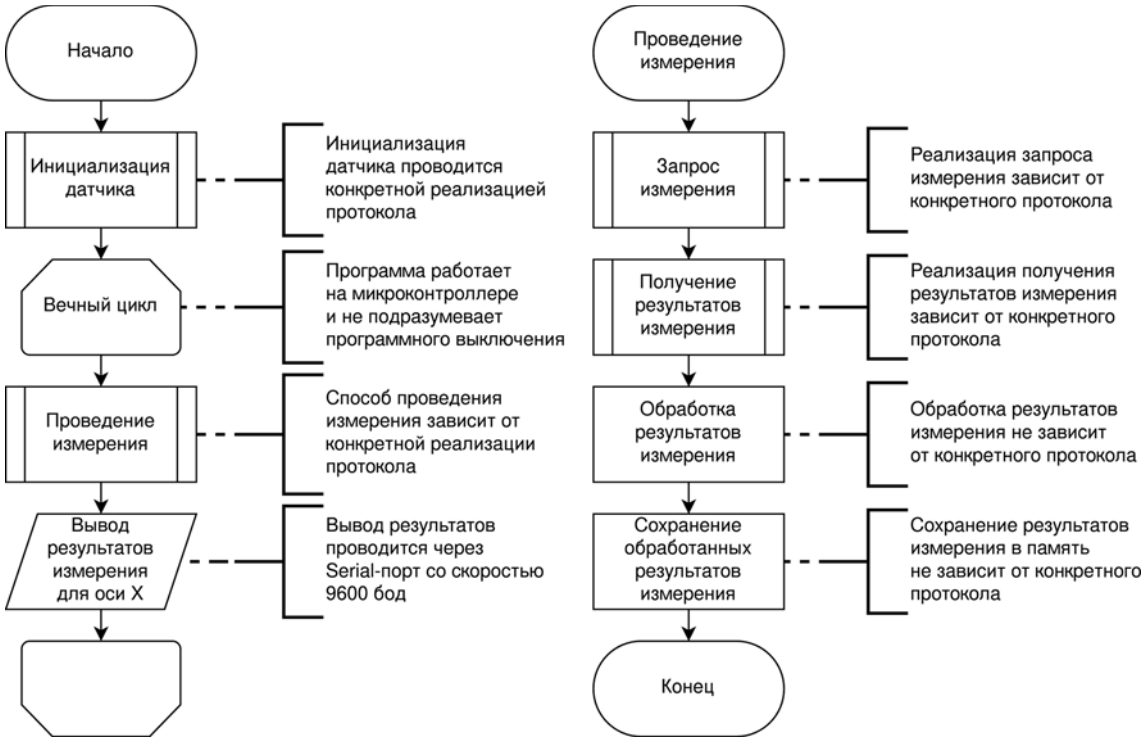

Рис. 1. Блок-схема программы обработки данных, получаемых от одного датчика

измерения, зависящая от реализации библиотеки. Все реализации процедуры измерения имеют сходный принцип работы: сначала происходит отправка запроса измерения на датчик, затем получение результатов измерения. Обе эти операции зависят от конкретного протокола передачи данных. Полученные данные затем обрабатываются и сохраняются в энергозависимую память для дальнейшего использования. Процедуры обработки и сохранения данных, в свою очередь, не зависят от протокола передачи данных.

Результаты эксперимента

В ходе эксперимента с помощью утилиты avr-size были проанализированы .elf-файлы скомпилированных программ. Программа, не обрабатывающая ни одного датчика, потребовала 3758 байт памяти программ и 151 байт памяти данных.

Результаты анализа скомпилированных программ обработки данных, получаемых от датчика, представлены в табл. 1 и 2, в скобках указан прирост по сравнению с программой, не обрабатывающей ни одного датчика.

Анализ результатов эксперимента

Использование библиотеки с независимыми реализациями протоколов приводит к наименьшим затратам по памяти как программы, так и данных. Вместе с тем использование такой библиотеки при работе с двумя датчиками приводит к приросту по памяти в 1856 байт (97 % относительно худшего результата). С точки же зрения качественных критериев библиотека с независимыми реализациями имеет перечисленные далее недостатки.

- Повторное использование кода практически невозможно, так как код с идентичной логикой находится в разных модулях и работает с разными протоколами.
- Внесение изменений в одну реализацию требует внесения таких же изменений во вторую реализацию.
- При внедрении кода в другую библиотеку возникает потребность в разработке двух версий библиотеки.

Таблица 1

Размер используемой памяти программ, байт

Реализация	SPI-датчик	I2C-датчик	I2C- и SPI-датчики
Независимые версии	8406 (4648)	8978 (5220)	10 262 (6504)
Шаблоны	8540 (4782)	9030 (5272)	10 442 (6684)
Интерфейс	8972 (5214)	9468 (5710)	10 640 (6882)

Таблица 2

Размер используемой памяти данных, байт

Реализация	SPI-датчик	I2C-датчик	I2C- и SPI-датчики
Независимые версии	292 (141)	394 (243)	416 (265)
Шаблоны	296 (145)	398 (247)	422 (271)
Интерфейс	318 (167)	420 (269)	466 (315)

Использование основанной на шаблонах библиотеки позволяет оценить средние результаты по требованиям к памяти. Такой подход дает наибольший прирост в 1902 байта (как худший результат, он принят за 100 %) при использовании двух протоколов одновременно. В аспекте качественных критериев библиотека, разработанная с использованием шаблонов, имеет следующие характеристики.

- Повторное использование кода возможно посредством шаблонов.
- Внесение изменений в реализацию обработки данных упрощено благодаря использованию шаблонов.
- При интегрировании в другую библиотеку возникает требование в применении шаблонов в новой библиотеке.

При использовании библиотеки, основанной на общем абстрактном классе, были получены худшие результаты по потреблению памяти. Однако использование этой библиотеки позволяет получить наименьший прирост по памяти при использовании двух протоколов в размере 1668 байт (87 % относительно худшего результата). Вместе с тем применение общего абстрактного класса обеспечивает выполнение следующих качественных критериев:

- повторное использование кода упрощено за счет вынесения логики в методы абстрактного класса;
- упрощено внесение изменений для функций, вынесенных в абстрактный класс или класс обработки данных;
- при внедрении кода в другую библиотеку на разработчика такой библиотеки не накладывается ограничений.

Прикладная интерпретация и иллюстрация результатов

На основе результатов эксперимента можно сделать перечисленные далее выводы.

- При разработке протоколонезависимого программного обеспечения для встраиваемой электроники с ограниченными ресурсами по памяти может быть выгодно разрабатывать независимые версии программного обеспечения.
- В случаях разработки программного обеспечения, способного работать на нескольких протоколах передачи данных одновременно, представляется разумным разделить логику передачи данных и логику обработки данных, особенно в случаях большого объема кодовой базы для обработки данных по сравнению с объемом кодовой базы для их передачи.

Модель экспериментов при использовании одного протокола представлена на рис. 2, при использовании двух протоколов — на рис. 3: *а* — использование двух реализаций библиотеки; *б* — использование шаблонов; *в* — использование общего интерфейса.

Как видно из иллюстраций, создание библиотеки без разделения ее на модули обеспечивает наименьший размер программы, а использование библиотеки с шаблонами приводит к тому, что компилятор генерирует код для каждой конкретной версии шаблона. Использование версии библиотеки с общим абстрактным классом дает наибольшие накладные расходы, которые, однако, возникают лишь один раз для всех протоколов.

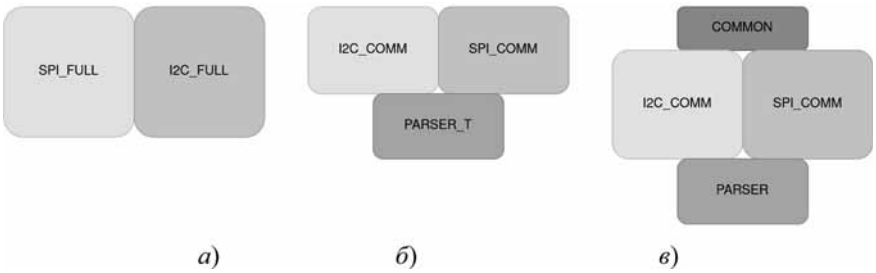

Рис. 2. Модель эксперимента с использованием одного протокола

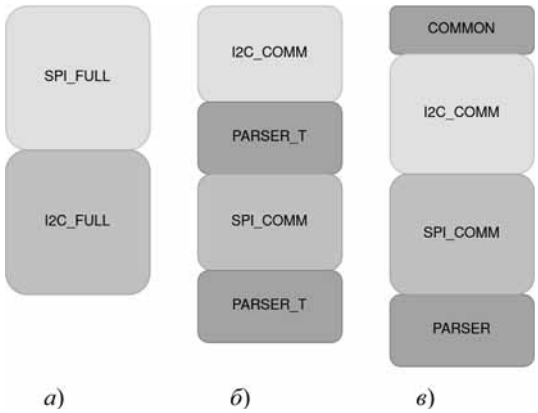

Рис. 3. Модель эксперимента с использованием двух протоколов

Полученные результаты согласуются с опытом, изложенным в работе [19], в частности, главах 7 и 11. Использование нескольких классов позволяет соблюсти принцип единственной ответственности и упростить долговременную поддержку программного обеспечения.

Заключение

В статье представлены подходы к разработке архитектуры протоколонезависимой библиотеки для обработки данных, получаемых от трехкомпонентного датчика силы. Датчик силы основан на трехосевом сенсоре магнитного поля MLX90393, способном передавать данные как с использованием SPI-, так и I2C-протоколов, в зависимости от исполнения датчика силы. Данные, передаваемые MLX90393, требуют дополнительной обработки перед использованием. Для решения этой задачи была разработана программная библиотека. Такая библиотека должна предоставлять инструмент обработки данных в зависимости от исполнения датчика.

Рассмотрены три альтернативные архитектуры библиотеки: два независимых программных модуля, отвечающих одновременно за обработку и передачу данных по I2C- или SPI-протоколу, два модуля, отвечающие за передачу данных, а также один, написанный с использованием шаблонов, модуль, отвечающий за их обработку, два связанных модуля для передачи данных, основанных на общем абстрактном классе, и модуль для обработки данных, работающий со ссылкой на реализацию абстрактного класса. Разработаны три экспериментальные реализации библиотеки, проведены девять экспериментов с различными вариантами использования библиотек, дополненные экспериментом без обработки данных для задания точки отсчета.

Предложены различные критерии оценки, как качественные, так и количественные, в том числе размер скомпилированной программы и простота интеграции в существующие библиотеки. Экспериментальные результаты проанализированы с использованием рассмотренных критериев, предложены практические рекомендации.

Список литературы

1. **Vorotnikov A., Bashevskaya O., Ilyukhin Y.** et al. Geometrical Approach for Industrial Robot Axis Calibration Using Laser Tracker // Proceedings of the 26th DAAAM International Symposium. — 2016. — P. 897—904.
2. **Воротников А. А., Подураев Ю. В., Ромаш Е. В.** Оценка погрешности определения центров вращения звеньев кинематической цепи для методики калибровки промышленных роботов // Измерительная техника. — 2015. — № 8. — С. 23—28.
3. **Ilyukhin Y. V., Poduraev Y. V.** Improving the precision of mechatronic robot drives // Russian Engineering Research. — 2016. — Vol. 36, No. 4. — P. 328—334.
4. **Илюхин Ю. В., Подураев Ю. В.** Повышение точности мехатронных приводов технологических роботов // СТИН. — 2015. — № 9. — С. 30—37.
5. **Янушевич О. О., Базикян Э. А., Чунихин А. А.** и др. Роботизированный мультифункциональный лазерный хирургический комплекс. Патент на изобретение RU 2693216 C1, 01.07.2019. Заявка № 2018119157 от 24.05.2018.
6. **Климов Д. Д., Воротников А. А., Соловьев М. А.** и др. Медицинская робототехника для нейрохирургии // Вестник МГТУ Станкин. — 2019. — № 1 (48). — С. 32—37.
7. **Буйнов М. А., Воротников А. А., Климов Д. Д.** и др. Роботические технологии в медицине и биопринтинге: состояние проблемы и современные тенденции // Вестник МГТУ Станкин. — 2017. — № 1 (40). — С. 127—131.
8. **Гречишников В. А., Исаев А. В., Пивкин П. М.** и др. Инструментальные решения для роботизированного фрезерования // Вестник МГТУ Станкин. — 2017. — № 4 (43). — С. 73—78.
9. **Гречишников В. А., Илюхин Ю. В., Исаев А. В.** и др. Повышение точности и производительности роботизированного фрезерования на основе траекторно-импедансного управления // Вестник МГТУ Станкин. — 2016. — № 4 (39). — С. 8—16.
10. **Гречишников В. А., Исаев А. В., Илюхин Ю. В.** и др. Концепция построения робототехнических комплексов для металлообработки и системы их инструментального обеспечения // Вестник МГТУ Станкин. — 2015. — № 4 (35). — С. 46—51.
11. **Базикян Э. А., Чунихин А. А., Янушевич О. О.** и др. Лазерная роботизированная медицинская установка. Патент на полезную модель RU 172817 U1, 25.07.2017. Заявка № 2017114622 от 27.04.2017.
12. **Мелешников А. М., Воротников А. А., Климов Д. Д., Подураев Ю. В.** Концепция прототипа рабочего органа робототехнической системы разрезания для определения контакта с десной в хирургической стоматологии // СТИН. — 2019. — № 8. — С. 31—33.
13. **Rezvani S., Nikolov N., Kim C.** et al. Development of a Vise with built-in Piezoelectric and Strain Gauge Sensors for Clamping and Cutting Force Measurements // Procedia Manufacturing. — 2020. — Vol. 48. — P. 1041—1046.
14. **Гойдин О. П., Кураев Е. В., Подураев Ю. В.** Робототехнический комплекс с силомоментным очувствлением для демонтажных работ // Вестник МГТУ Станкин. — 2015. — № 1 (32). — С. 78—82.
15. **Forte M., Correia W., Nogueira F., Torrico B.** Reference Tracking of a Nonholonomic Mobile Robot using Sensor Fusion Techniques and Linear Control // IFAC-PapersOnLine. — 2018. — Vol. 51, No. 4. P. — 364—369.
16. **Colomina I., Molina P.** Unmanned aerial systems for photogrammetry and remote sensing: A review // ISPRS Journal of Photogrammetry and Remote Sensing. — 2014. — Vol. 92. — P. 79—97.
17. **Akbari H., Kazerooni A.** Improving the coupling errors of a Maltese cross-beams type six-axis force/moment sensor using numerical shape-optimization technique // Measurement. — 2018. — Vol. 126. — P. 342—355.
18. **Ченцов П.** Об одном подходе к разработке кроссплатформенных приложений на языке C++ // Программная инженерия. — 2019. — Т. 10, № 3. — С. 105—113.
19. **Martin R.** Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition, Prentice Hall, 2016. 430 p.

Development of Protocol-Independent Modular Architecture for Three Axis Force Sensor Data Processing

P. A. Trunin, patrunin@mie.hse.ru, **L. S. Prokhorenko,** pro.leonid@gmail.com,
D. S. Mishchenkov, dmish32@gmail.com, **D. D. Klimov,** daniil.klimov@gmail.com,
MSTU "STANKIN", Moscow, 127055, Russian Federation

Corresponding author:

Prokhorenko Leonid S., Junior Researcher, MSTU "STANKIN", Moscow, 127055, Russian Federation
E-mail: pro.leonid@gmail.com

Received on September 15, 2020
Accepted on October 22, 2020

The protocol-independent architecture for three axis force sensor data processing is considered. The force sensor is based on a MLX90393 triaxial magnetic sensor, capable of transmitting data via SPI protocol or I2C protocol, depending on the force sensor modification. The data generated by the MLX90393 requires additional processing before usage. To address this issue a software library is developed. Such a library shall provide means for protocol selection based on the force sensor modification. Three alternative architectural library implementations are considered: the two independent program modules for both SPI and I2C protocol data transmission and processing, the two dependent program modules for SPI and I2C data transmission and one template-based module for the data processing, the two dependent program modules for SPI and I2C data transmission based on a single abstract class and the data processing module working with a pointer to the abstract class implementation. Three experimental library implementations are developed, nine experiments are performed with varying library usage setups, supplemented with one experiment with no data processing for the reference. Various criteria are considered, both quantitative and qualitative, including, but not limited to the compiled program size and the ease of library integration. The experimental results are analysed with the help of stated criteria, conclusions and practical recommendations are made.

Keywords: *software architecture, software development, development practices, maintainable software, integrated systems*

For citation:

Trunin P. A., Prokhorenko L. S., Mishchenkov D. S., Klimov D. D. Development of Protocol-Independent Modular Architecture for Three Axis Force Sensor Data Processing, *Programmnaya Ingeneria,* 2021, vol. 12, no. 1, pp. 3-7

DOI: 10.17587/prin.12.3-7

References

1. **Vorotnikov A., Bashevskaya O., Ilyukhin Y., Romash E., Isaev A., Poduraev Y.** Geometrical Approach for Industrial Robot Axis Calibration Using Laser Tracker, *Proceedings of the 26th DAAAM International Symposium*, 2016, Vienna, Austria, 2016, pp. 897—904.

2. **Vorotnikov A., Poduraev Y., Romash E.** Estimation of the error in determining the centers of rotation of the links of the kinematic chain for the calibration method of industrial robots, *Izmeritelnaya teknika*, 2015, no. 8, pp. 23—28 (in Russian).

3. **Ilyukhin Y., Poduraev Y.** Improving the precision of mechatronic robot drives, *Russian Engineering Research*, 2016, vol. 36, no. 4, pp. 328—334.

4. **Ilyukhin Y., Poduraev Y.** Improving the precision of mechatronic robot drives, *STIN*, 2015, no. 9, pp. 30—37 (in Russian).

5. **Yanushevich O., Basikyan E., Chunichin A., Vorotnikov A., Klimov D., Poduraev Y.** Robotic multifunctional laser surgical complex. Patent RU 2693216 C1, 01.07.2019. Application № 2018119157 24.05.2018 (in Russian).

6. **Klimov D., Vorotnikov A., Soloviev M., Poduraev Y., Grin A., Krilov V.** Medical robotics for neurosurgery, *Vestnik MGTU Stankin*, 2019, no. 1 (48), pp. 32—37 (in Russian).

7. **Buynov M., Vorotnikov A., Klimov D., Malishev I., Mironov V., Parfenov V., Pereyra F. D., Poduraev Y., Hesuani Y.** Robotic technologies in medicine and bioprinting: state of the art and current trends, *Vestnik MGTU Stankin,* 2017, no. 1 (40), pp. 127—131 (in Russian).

8. **Grechishnikov V., Isaev A., Pivkin P., Ilyukhin Y., Vorotnikov A., Kolisnechenko R., Byanki D., Leonesio M**. Tooling solutions for robotic milling, *Vestnik MGTU Stankin,* 2017, no. 4 (43), pp. 73—78 (in Russian).

9. **Grechishnikov V., Ilyukhin Y., Isaev A., Kolisnechenko R., Pivkin P., Vorotnikov A., Byanki D., Pedrocchi N.** Improving the accuracy and productivity of robotic milling based on path-impedance control, *Vestnik MGTU Stankin,* 2016, no. 4 (39), pp. 8—16 (in Russian).

10. **Grechishnikov V., Isaev A., Ilyukhin Y., Pivkin P., Vorotnikov A., Harchenko A., Byanki D., Leonesio M., Pedrocchi N., Tosatti L.** The concept of building robotic systems for metalworking and their instrumental support, *Vestnik MGTU Stankin,* 2015, no. 4 (35), pp. 46—51 (in Russian).

11. **Basikyan E., Chunichin A., Yanushevich O., Poduraev Y., Buynov M., Klimov D., Vorotnikov A.** Laser robotic medical device. Patent RU 172817 U1, 25.07.2017. Application № 2017114622 27.04.2017 (in Russian).

12. **Meleshnikov A., Vorotnikov A., Klimov D., Poduraev Y.** The concept of a prototype of robotic cutting system tool for contact with the gum determination in dental surgery, *STIN*, 2019, no. 8, pp. 31—33 (in Russian).

13. **Rezvani S., Nikolov N., Kim C., S. Park S., Lee J.** Development of a Vise with built-in Piezoelectric and Strain Gauge Sensors for Clamping and Cutting Force Measurements, *Procedia Manufacturing,* 2020, vol. 48, pp. 1041—1046.

14. **Goidin O., Kuraev E., Poduraev Y.** Robotic complex with force-torque sensing for dismantling works, *Vestnik MGTU Stankin*, 2015, no. 1 (32), pp. 78—82 (in Russian).

15. **Forte M., Correia W., Nogueira F., Torrico B.** Reference Tracking of a Nonholonomic Mobile Robot using Sensor Fusion Techniques and Linear Control, *IFAC-PapersOnLine*, 2018, vol. 51, no. 4, pp. 364—369.

16. **Colomina P., Molina P.** Unmanned aerial systems for photogrammetry and remote sensing: A review, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2014, vol. 92, pp. 79—97.

17. **Akbari H., Kazerooni A.** Improving the coupling errors of a Maltese cross-beams type six-axis force/moment sensor using numerical shape-optimization technique, *Measurement*, 2018, vol. 126, pp. 342—355.

18. **Chentsov P.** On one approach to developing cross-platform C++ applications, *Programnaya Ingeneria,* 2019, vol. 10, no. 3, pp. 105—113 (in Russian).

19. **Martin R.** *Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series),* 1st Edition, Prentice Hall, 2016, 430 p.

УДК 004.056.55 DOI: 10.17587/prin.12.8-20

Е. А. Курганов, вед. разработчик ПО, kuev@yandex.ru, ФГБУ "НИИ Восход", Москва

Об аппаратной реализации сбалансированных S-блоков

Представлены результаты исследований эффективности универсальных алгоритмов синтеза аппаратных схем (наивный на основе СДНФ, Шеннона, Лупанова, упрощения ДНФ) применительно к сбалансированным S-блокам. Описана авторская программа, позволяющая получить схемную реализацию S-блока на языке Verilog. Для каждого алгоритма приведена оценка глубины и сложности реализации S-блока, полученной с помощью программы при произвольном n. Приведен алгоритм минимизации произвольной системы булевых функций, предложенный автором. Дано сравнение эффективности всех рассмотренных алгоритмов при реализации S-блоков из криптографических стандартов.

Ключевые слова: *S-блок, аппаратная реализация, оптимизация глубины схем, оптимизация сложности схем, потоковые шифры, блочные шифры*

Введение

S-блок — это нелинейное преобразование, принимающее на вход n бит и возвращающее m бит. Такое преобразование проще всего представлять как таблицу подстановок размером $n \times m$. Чаще всего в криптографии используют только сбалансированные S-блоки (это значит, что отображение, задаваемое S-блоком, является биекцией). Иными словами, это значит, что число входных битов равно числу выходных битов.

S-блок — важная часть большинства симметричных шифров. Подбор правильной подстановки позволяет сделать связь между ключом и зашифрованным текстом более сложной (нелинейной), что значительно усложняет взлом.

В настоящей работе рассматривается аппаратная реализация сбалансированных S-блоков. Определяющими параметрами производительности таких реализаций являются глубина и сложность схемы. Под глубиной понимается длина максимального простого пути, а под сложностью — общее число элементов схемы. Рассматривается базис из элементов конъюнкции, дизъюнкции, отрицания и задержки. При этом отрицание игнорируется при вычислении глубины и сложности (вычисление проводится по тем же правилам, что и в работах [1—3]).

Для того чтобы сравнить эффективность аппаратных реализаций S-блоков, полученных разными методами, автором была написана программа, позволяющая построить схему на языке Verilog для заданной подстановки тем или иным методом синтеза. В начале статьи дано описание программы: кратко описан интерфейс, а затем указано подробное описание всех реализованных методов синтеза. Среди них есть новый алгоритм, разработанный автором. Для него доказывается корректность. Далее приведена оценка глубины и сложности схем, получаемых с помощью программы каждым из методов. В заключение приведено сравнение всех перечисленных методов в плане сложности и глубины схем S-блоков, которые используются в популярных блочных и потоковых симметричных шифрах.

Программа для генерации логических схем

Описание интерфейса. Программа написана на языке C++ и рассчитана на S-блоки размером не больше, чем 16×16 бит. На вход программы подается сама подстановка (записанная в файл) и метод, которым надо сгенерировать схему. На выходе получается файл со схемой на языке Verilog.

Поддерживаются следующие шесть методов синтеза:

— наивный;
— улучшенный наивный;
— Шеннона;
— Лупанова;
— на основе упрощения дизъюнктивной нормальной формы (ДНФ);
— новый авторский метод.

Описание и исходный код программы можно найти по адресу: https://github.com/ekurganov/sbox_circuit.

Рассмотрим более подробно, как идет построение схемы каждым из указанных выше методов. Для начала рассмотрим построение более простых частей схемы, которые используются в нескольких методах. Оценки глубины и сложности для предложенных схем приведены ниже.

Дешифратор. ***Определение 1.*** *Дешифратор* от переменных x_1, ..., x_n — схема, реализующая все конъюнкции вида $x_1^{\sigma_1} \& x_2^{\sigma_2} \& \ldots \& x_n^{\sigma_n}$, где $\sigma_i = 0, 1$; $i = 1, \ldots, n$; $x^1 = x$, $x^0 = \overline{x}$ (всего будет 2^n конъюнкций, на каждом наборе только одна из них будет равна 1). Обозначим эту схему D_n.

Приведем способ построения дешифратора от n переменных. Сделаем это по индукции.

База. При $n = 1$ дешифратор тривиален и реализуется с глубиной и сложностью 0 (рис. 1); при

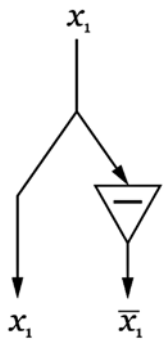

Рис. 1. Схема дешифратора D_1

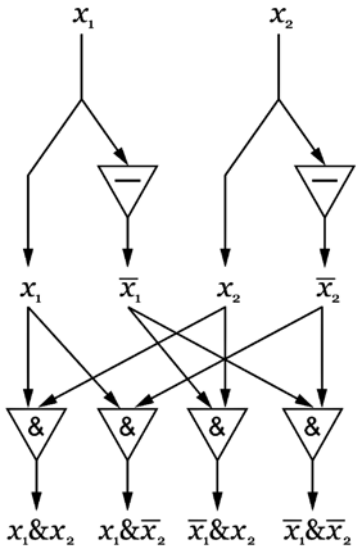

Рис. 2. Схема дешифратора D_2

$n = 2$ дешифратор можно реализовать с помощью схемы, изображенной на рис. 2

Индуктивный переход. Пусть дешифратор построен для всех $i < n$. Заметим, что для всех $n \in \mathbb{N}$ верно следующее равенство: $n = \left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor$.

Тогда дешифратор для $i = n$ можно получить, воспользовавшись схемой, приведенной на рис. 3. В данном случае $t = \left\lceil \frac{n}{2} \right\rceil$.

Все булевы функции. Обозначим схему, которая реализует все булевы функции от переменных $x_1, ..., x_n$, через Φ_n. Покажем, как строится данная схема. Она состоит из двух частей. Первая часть — это дешифратор D_n. Вторую часть можно разделить на несколько слоев, перечисленных далее.

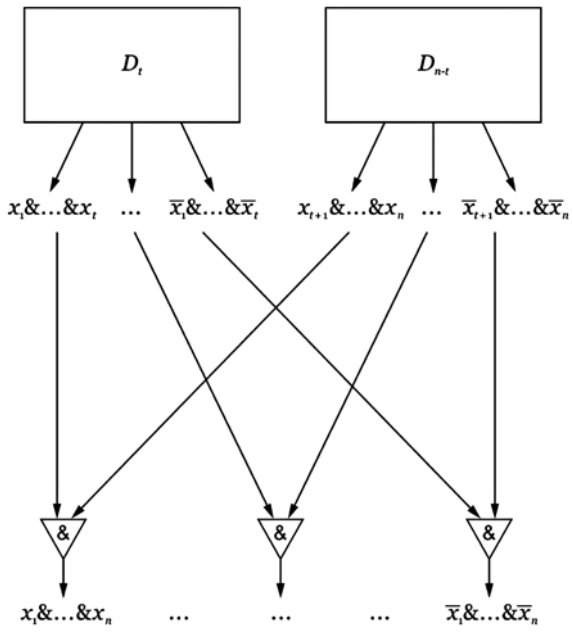

Рис. 3. Рекурсивное построение дешифратора D_n

- Первый слой — выход дешифратора, 2^n элементарных конъюнкций; назовем их $K_1, ..., K_{2^n}$.
- Второй слой — все возможные комбинации из двух элементарных конъюнкций $K_i \vee K_j$, $i, j, ..., 1, ..., 2^n$, $i \neq j$.
- Третий слой — все возможные комбинации из трех и четырех элементарных конъюнкций $K_i \vee K_j \vee K_k$, $K_i \vee K_j \vee K_k \vee K_l$, $i, j, k, l = 1, ..., 2^n$; i, j, k, l попарно различны. Для вычисления четверок используются двойки $K_i \vee K_j$, $i, j = 1, ..., 2^n$, $i \neq j$, вычисленные на втором слое; для вычисления троек используются двойки из второго слоя и элементарные конъюнкции из первого слоя.
- Слой d, $d < n$ — все возможные комбинации (дизъюнкции) из $2^{d-2} + 1, ..., 2^{d-1}$ конъюнкций K_i с попарно различными индексами. Для вычисления используются все результаты, полученные на предыдущих слоях.
- Слой n, последний. На этом слое достаточно вычислить все возможные комбинации из $2^{n-2} + 1, ..., 2^{n-1} - 1$ и половину комбинаций из 2^{n-1} конъюнкций K_i. Остальные функции можно получить, добавив отрицание на вычисленные ранее функции. Также стоит учитывать, что n функций, принимающие значение 1 ровно на половине наборов, являются селекторами (т. е. равны x_i, $i = 1, ..., n$) и не требуют реализации. Следовательно, итоговую сложность схемы можно дополнительно уменьшить на n.

Таким образом, на первом слое реализованы константы и все функции от n переменных, принимающие значение 1 только на одном наборе. На втором слое реализованы все функции от n переменных, равные 1 только на двух наборах, на третьем слое — на трех и четырех наборах, и т. д. Достаточно вычислить половину всех функций, так как другую половину можно получить, добавив отрицание (оно не учитывается при учете глубины и сложности).

Дизъюнкция всех переменных. Обозначим через AF_n схему, которая реализует все возможные дизъюнкции элементов $x_1, ..., x_n$.

Данная схема строится таким же способом, как и вторая часть схемы Φ_n. Однако в данном случае нельзя воспользоваться тем обстоятельством, что достаточно посчитать лишь половину дизъюнкций, а другую половину можно получить с помощью отрицания. Поэтому к конструкции из схемы Φ_n добавится еще один слой, на котором будут считаться дизъюнкции элементов, равных 1 на позициях $2^{n-1} + 1, ..., 2^n$.

Обзор реализованных методов синтеза

Прежде всего следует отметить, что имеет место следующее утверждение.

Утверждение 1. Для любого сбалансированного S-блока размером $n \times n$ бит верно следующее:

- каждая булева функция, реализующая выходной бит S-блока, равна 0 и 1 на 2^{n-1} наборах переменных;

- обе булевы функции из любой пары, реализующей выходные биты S-блока, одновременно равны 1 на 2^{n-2} наборах переменных.

Данное утверждение является частным случаем критерия биективности отображения, задаваемого n булевыми функциями от n переменных [4].

Наивный метод на основе совершенной ДНФ (СДНФ). При использовании данного метода схема для сбалансированного S-блока от n переменных строится следующим образом:

1) строится схема дешифратора D_n;

2) для каждого выхода S-блока y_j, $j = 1, ..., n$ выбирается 2^{n-1} из 2^n получившихся на прошлом шаге конъюнкций, которым соответствуют те входные значения, на которых $y_j = 1$;

3) для каждого выхода S-блока y_j, $j = 1, ..., n$ считается дизъюнкция 2^{n-1} элементов, выбранных на прошлом шаге.

Улучшенный наивный метод. Описанный выше наивный метод реализации S-блоков можно улучшить, воспользовавшись утверждением 1. Для этого сгруппируем выходы дешифратора D_n. Разобьем выходы S-блока $z_1, ..., z_n$ на пары произвольным образом. Для каждой пары (j_1, j_2) выберем те выходы дешифратора, на которых $z_{j_1} = z_{j_2} = 1$. Таких элементов будет ровно $2^{n-1} + 2^{n-2}$, причем 2^{n-2} из них будут общими. Пусть для z_{j_1} — это будут элементы $y_1^{j_1}, ..., y_{2^{n-1}}^{j_1}$, а для z_{j_2} — $y_1^{j_2}, ..., y_{2^{n-1}}^{j_2}$, и при этом $y_{2^{n-2}+1}^{j_1} = y_{2^{n-2}+1}^{j_2}, ..., y_{2^{n-1}}^{j_1} = y_{2^{n-1}}^{j_2}$. Тогда для данной пары можно сначала параллельно посчитать дизъюнкции $\vee_{i=1}^{2^{n-2}} y_i^{j_1}$, $\vee_{i=1}^{2^{n-2}} y_i^{j_2}$, $\vee_{i=2^{n-2}+1}^{2^{n-1}} y_i^{j_1}$, а потом уже дизъюнкцию двух посчитанных элементов для каждого выхода. Это позволит уменьшить сложность для каждой пары на 2^{n-2}.

Метод на основе упрощения ДНФ. Рассмотрим каждый выходной бит S-блока $n \times n$ как отдельную булеву функцию от n переменных.

Применим для каждой из этих функций алгоритм ESPRESSO [5], который считает ДНФ, близкую к минимальной (для небольших n она совпадает с минимальной). Далее для полученной ДНФ схема строится таким же образом, как и в методе на основе СДНФ. А именно, сначала считаются все необходимые элементарные конъюнкции, после чего считается их дизъюнкция.

Данный алгоритм позволяет получить схему, глубина которой не больше, чем глубина простой реализации S-блоков, приведенной выше. Действительно, так как данный алгоритм позволяет получить одну из тупиковых ДНФ, которая по своему построению будет иметь не больше дизъюнктов, чем СДНФ, то и глубина полученной схемы будет не больше, чем глубина простой реализации.

Сложность такой схемы можно уменьшить следующим способом: начиная с числа $t = n - 1$ и уменьшая его до 2, начнем искать среди каждой элементарной конъюнкции, входящей в ДНФ какого-либо из выходов S-блока, общие конъюнкции, состоящие ровно из t элементов. При нахождении таких конъюнкций будем предвычислять их до вычисления значения функции выхода таким образом, чтобы общая глубина вычисления исходной конъюнкции не увеличилась. На каждом новом шаге будем искать общие конъюнкции также среди найденных ранее общих частей. Таким образом, итоговая схема будет иметь примерно такой вид:

- общие элементы вида $x_{i_1} \,\&\, x_{i_2}$;
- общие элементы вида $x_{i_1} \,\&\, x_{i_2} \,\&\, x_{i_3}$;
- ...
- общие элементы вида $x_{i_1} \,\&\, ... \,\&\, x_{i_{n-1}}$;
- $z_1 = \vee x_{i_1}^{\sigma_{i_1}} \,\&\, ... \,\&\, x_{i_k}^{\sigma_{i_k}}$;
- ...
- $z_n = \vee x_{i_1}^{\sigma_{i_1}} \,\&\, ... \,\&\, x_{i_k}^{\sigma_{i_k}}$.

Метод Шеннона. Данный метод устроен так. Вводится параметр k, $0 < k < n$ и рассматривается разложение

$$f(x_1, ..., x_n) = \bigvee_{(\sigma_{k+1}, ..., \sigma_n)} x_{k+1}^{\sigma_{k+1}} ... \\ ... \, x_n^{\sigma_n} f(x_1, ..., x_k, \sigma_{k+1}, ..., \sigma_n). \quad (1)$$

Пользуясь формулой (1), можно построить схему для функции $f(x_1, ..., x_n)$ следующим образом.

Шаг 1. Выбирается такое k, при котором сложность схемы имеет минимальное значение.

Шаг 2. Для первых k переменных реализуются все булевы функции от k аргументов (строится схема для Φ_k).

Шаг 3. Параллельно с шагом 2 для остальных $n - k$ переменных строится схема дешифратора D_{n-k}.

Шаг 4. Для каждой из 2^{n-k} конъюнкций (выход D_{n-k}) второй группы делается конъюнкция с необходимой функцией $f(x_1, ..., x_k)$ из первой группы (см. формулу (1)).

Шаг 5. Вычисляется дизъюнкция всех элементов, вычисленных на предыдущем шаге (см. формулу (1)).

Подробнее о данном методе можно узнать в работе [6].

Метод Лупанова. Метод Лупанова похож на метод Шеннона. Переменные также делятся на две группы (параметр k). Отличие состоит в том, что на первом шаге вместо всех функций от k переменных строится некоторое их подмножество.

Более подробно: булева функция представляется в виде таблицы размером $2^k \times 2^{n-k}$; строки индексируются значениями первых k переменных, столбцы — значениями оставшихся $n - k$; таким образом, на пересечении столбца и строки находится значение функции для соответствующего набора аргументов.

Данная таблица делится на горизонтальные полосы шириной s, $1 < s < 2^k$ (последняя полоса, возможно, будет короче остальных; ее ширину обозначим s', $s' \leqslant s$). Полученные полосы нумеруются сверху вниз числами от 1 до p, где $p = \left\lceil \frac{2^k}{s} \right\rceil$.

При отдельном рассмотрении некоторой полосы можно заметить, что среди ее столбцов при небольшом s будет много повторений; про одинаковые столбцы, находящиеся в одной полосе, говорят, что они одного сорта.

Число сортов для полосы с номером i обозначается $t(i)$. Понятно, что для любой полосы $t(i) \leqslant 2^s$ (для последней $t(p) \leqslant 2^{s'}$).

Пусть для некоторого i:

- $(\gamma_1, ..., \gamma_s)$ — столбец i-й строки j-го сорта;
- $(\sigma_1^l, ..., \sigma_k^l)$ — аргументы функции, соответствующие l-й строке i-й полосы.

Рассмотрим булеву функцию

$$f_{ij}(x_1, ..., x_k) =$$

$$= \begin{cases} \gamma_l, & \text{если} (\sigma_1, ..., \sigma_k) = (\sigma_1^l, ..., \sigma_k^l),\ l = 1, ..., s; \\ 0, & \text{если} (\sigma_1, ..., \sigma_k) \text{ не принадлежит } i\text{-й полосе.} \end{cases}$$

Так как исходный столбец $(\sigma_{k+1}, ..., \sigma_n)$ состоит из p коротких столбцов, на который его делят полосы, то верно следующее разложение:

$$f(x_1, ..., x_k, \sigma_{k+1}, ..., \sigma_n) = \bigvee_{i=1}^{p} f_{ij_i}(x_1, ..., x_k), \qquad (2)$$

где j_i — номер сорта соответствующего короткого столбца, принадлежащего i-й полосе.

Схема по данному разложению строится следующим образом.

Шаг 1. Выбираются такие s и k, при которых сложность схемы имеет минимальное значение.

Шаг 2. Строится схема дешифратора D_k.

Шаг 3. По СДНФ реализуются все $f_{ij}(x_1, ..., x_k)$, $i = 1, ..., p$, $j = 1, ..., t(i)$.

Шаг 4. В соответствии с разложением (2) реализуется функция $f(x_1, ..., x_k, \sigma_{k+1}, ..., \sigma_n)$.

Шаг 5. Параллельно с шагами 1—3 реализуется схема D_{n-k}.

Шаг 6. Для каждой из 2^{n-k} конъюнкций (выход D_{n-k}) второй группы делается конъюнкция с необходимой функцией $f(x_1, ..., x_k)$ из первой группы.

Шаг 7. Вычисляется дизъюнкция всех элементов, вычисленных на предыдущем шаге.

Подробнее о данном методе можно узнать в работе [6].

Алгоритм минимизации сложности системы булевых функций

Представим новый, разработанный автором алгоритм, который можно применять не только для S-блоков, но и для произвольных систем булевых функций.

Рассмотрим систему из m булевых функций от n переменных $f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)$. Каждой из этих функций можно поставить в соответствие ее вектор значений $p_1, ..., p_m$ длины 2^n, где $p_{ij} \in \{0, 1\}$, $i = 1, ..., m$; $j = 1, ..., 2^n$. Причем $p_{ij} = 1$ тогда и только тогда, когда в таблице истинности функции f_i на j-м месте стоит 1. Будем рассматривать только системы из попарно различных векторов.

Введем обозначение: для каждого вектора p_i, $i = 1, ..., m$:

$$N_\vee(p_i) = \begin{cases} wt_H(p_i) - 1, & wt_H(p_i) > 1, \\ 0, & wt_H(p_i) \leqslant 1, \end{cases}$$

где $wt_H(v)$ означает число координат, равных единице (так называемый вес Хэмминга), для вектора v.

Для пары векторов p_i, p_j, $i, j = 1, ..., m$, $i \neq j$, определим вектор $p_i \,\&\, p_j$ следующим образом: $(p_i \,\&\, p_j)_k = p_{i_k} \,\&\, p_{j_k}$.

Основная идея алгоритма состоит в следующем.

Шаг 1. Разбить исходное множество векторов на пары $(p_i, p_j)_t$, $t = 1, ..., n/2$ так, что $N_\vee(v_t = (p_i \,\&\, p_j)_t)$ максимально.

Шаг 2. Для каждой пары $(p_i \,\&\, p_j)_t$ (где l — номер шага алгоритма):

- добавить в изначально пустое множество $P_{l,2}$ векторы $p_i \,\&\, \overline{v_t}$; $p_j \,\&\, \overline{v_t}$.
- добавить в изначально пустое множество $P_{l,3}$ вектор v_t.

Шаг 3. Повторить шаги 1—2 для множества $P_{l,2}$ до тех пор, пока $N_\vee(v_t) > 0$.

Шаг 4. Повторить шаги 1—2 для множества $P_{l,3}$ до тех пор, пока $N_\vee(v_t) > 0$.

Описанный процесс легко представить в виде построения бинарного дерева. Оно строится по следующим правилам:

- корень дерева — исходное множество векторов $p_1, ..., p_m$, разбитое на пары;
- левый потомок данного узла — множество $P_{l,2}$;
- правый потомок данного узла — множество $P_{l,3}$;
- узел не имеет потомков, если содержит только один вектор или $N_\vee(v_t) = 0 \,\forall\, t = 1, ..., n/2$.

Более подробно рекурсивный алгоритм построения дерева выглядит так. Для каждого узла, начиная с корня, выполняется процедура, описанная в алгоритме 1. Похожий подход для построения аппаратной реализации системы булевых функций используется в статье [7].

Утверждение 2. Обозначим $H = \min\{2m-1, 2^{n-1}\}$. Тогда данный алгоритм имеет сложность $O(m^2 2^{H+n-1})$.

Алгоритм 1. Построение бинарного дерева

```
1:  procedure Separate (node, m)
2:    P2 = P3 = ∅
3:   flag = 0
4:   if m = 1 then
5:    return
6:   for t ← 1, m/2 do
7:     найти пару векторов p_i, p_j с максимальным N_∨
8:     if N_∨ > 0 then
9:     if flag = 0 then
10:      flag ← 1
11:    comm ← p_i & p_j
12:    dif1 ← p_i & ¬comm
13:    dif2 ← p_j & ¬comm
14:    P2 = P2 ∪ dif1 ∪ dif2
```

```
15:   P3 = P3 ∪ comm
16:  if flag = 1 then
17:    добавить левого потомка left с множеством
       векторов P2
18:    Separate (left, m)
19:    добавить правого потомка right с множеством
       векторов P3
20:    Separate (right, m/2)
21:  else
22:   return
```

Доказательство. Операции, выполняемые для каждого узла дерева, можно разбить на следующие два этапа.

- Поиск пары векторов с максимальным общим числом единиц: всего проверяется $\frac{m}{2}(m-1)$ пар векторов. Для каждой пары проводится два сравнения по каждой из 2^n координат (сравнение с единицей и между собой). Всего получается $m(m-1)2^n$ сравнений.
- Вычисление для каждой пары векторов общих единиц и разницы (всего три вектора, см. алгоритм 1). Таким образом, вычисляется $3\frac{m}{2}2^n = 3m2^{n-1}$ конъюнкций.

Всего получается не больше, чем $2^n\left(m^2 - m + \frac{3m}{2}\right) = m\left(m + \frac{1}{2}\right)2^n$ операций для одного узла.

Теперь оценим максимально возможное число узлов в дереве. Для этого оценим максимально возможную высоту дерева. Всего в левом потомке $C_m^2 = \frac{m}{2}(m-1)$ различных пар векторов. Если на i-м шаге алгоритма в крайней левой листовой вершине была выбрана пара векторов p_{i_1}, p_{i_2} с числом общих единиц c_{i_1}, то на $(i+1)$-м шаге у данной пары векторов не будет общих единиц. Левая ветвь перестанет расти, когда для любой пары векторов, соответствующих крайнему левому потомку, число общих единиц будет равно нулю. В общем случае это произойдет после перебора всех возможных пар векторов.

Во время применения алгоритма 1 к одному узлу можно взять $\left\lfloor \frac{m}{2} \right\rfloor$ пар. Следовательно, для полного перебора понадобится не меньше, чем $\frac{m}{2}(m-1)\Big/\left\lfloor \frac{m}{2} \right\rfloor$ итераций. Однако перебрать все пары векторов ровно за указанное число итераций получается не всегда.

Для пояснения рассмотрим пример. Пусть $m = 6$. Всего получается $3 \cdot 5 = 15$ пар. Следовательно, минимальное число итераций, за которое можно перебрать все пары, составляет 5. Приведем выбор пар, который позволяет это сделать:

12	34	56
13	25	46
14	26	35
15	24	36
16	23	45

Однако если выбирать пары в другом порядке, понадобится больше итераций. Представим пример, когда перебор всех пар занимает 7 итераций:

12	34	56
13	24	56
14	23	56
15	26	34
16	25	34
35	46	12
36	45	12

Получается, что число итераций, необходимое для перебора всех пар векторов, зависит от того, какие пары выбираются вначале. В алгоритме 1 пары векторов нельзя выбирать в произвольном порядке, так как каждый раз ищется пара с максимальным числом общих единиц. Поэтому нужно оценить сверху число итераций, необходимое для перебора всех пар векторов при любом выборе первых пар. Для того чтобы сделать это, докажем две леммы.

Лемма 1. Пусть $m = 2^k$, $k \in \mathbb{N}$, и за один раз можно взять 2^{k-1} пар векторов. Тогда вне зависимости от того, какие пары будут выбраны вначале, все пары можно перебрать за $2^k - 1$ итераций.

Доказательство. Докажем данное утверждение индукцией по k.

База индукции. Для $k = 1$ есть всего одна пара. Ее можно рассмотреть за одну итерацию.

Индуктивный переход. Пусть утверждение верно для $k = t$. Докажем, что оно верно для $k = t + 1$. Разобьем исходное множество чисел $A = \{1, \ldots, 2^{t+1}\}$ на два подмножества: $A_1 = \{1, \ldots, 2^t\}$ и $A_2 = \{2^t + 1, \ldots, 2^{t+1}\}$. Для каждого из этих множеств по предположению индукции верно, что все возможные пары чисел из них можно перебрать за $2^t - 1$ итераций вне зависимости от начального выбора пар.

Теперь осталось перебрать все пары вида a_1a_2, где $a_1 \in A_1$, $a_2 \in A_2$. Это можно сделать ровно за 2^t итераций, так как мощность левого множества равна мощности правого множества и, следовательно, выбор первой, второй и т. д. пары не ограничивает выбор последней пары.

Таким образом, все пары можно перебрать за $2^t - 1 + 2^t = 2^{t+1} - 1$. Лемма доказана.

Лемма 2. Пусть $m = 2^k + l$, $1 \leqslant l < 2^k$, $k \in \mathbb{N}$, и за один раз можно взять $\left\lfloor \frac{m}{2} \right\rfloor$ пар векторов. Тогда вне зависимости от того, какие пары будут выбраны вначале, все пары можно перебрать не более, чем за $2m - 1$ итераций.

Доказательство. Докажем данное утверждение индукцией по k.

База индукции. Для $k = 1$ l может быть равно только единице. Следовательно, $m = 3$. Тогда всего надо перебрать $\frac{3 \cdot 2}{2} = 3$ пары векторов. Сделать это можно за 3 итерации. $3 < 3 \cdot 2 - 1 = 5$.

Индуктивный переход. Пусть утверждение верно для $k = t - 1$. Докажем, что оно верно для $k = t$. Разо-

бьем исходное множество чисел $A=\{1, ..., 2^t + l\}$ на два подмножества: $A_1 = \{1, ..., 2^t\}$; $A_2 = \{2^t + 1, ..., l\}$.

Перебрать все пары в подмножестве A_1 вне зависимости от выбора первых пар по лемме 1 можно за $2^t - 1$ итераций. Для подмножества A_2 сделать то же самое можно не более, чем за $2l - 1$ итераций по предположению индукции.

Теперь осталось перебрать все пары вида a_1a_2, где $a_1 \in A_1$, $a_2 \in A_2$. Это можно сделать не более, чем за $l\left\lceil 2^t/l \right\rceil$ итераций, так как мощности множеств не равны, и, следовательно, выбор первой, второй и т. д. пары может ограничить выбор последней пары.

Таким образом, все пары можно перебрать не более, чем за

$$\max\{2l - 1,\ 2^t - 1\} + l\left\lceil \frac{2^t}{l} \right\rceil < \max\{2l - 1,\ 2^t - 1\} + 2^t + l.$$

Для уточнения оценки рассмотрим отдельно два случая:

- $l \leqslant 2^{t-1}$; тогда необходимое число итераций не превосходит $2^t - 1 + 2^t + l = 2^{t+1} + l - 1 < 2m - 1$;
- $l > 2^{t-1}$; тогда необходимое число итераций не превосходит

$$2l - 1 + 2^t + l = 2^t + 2l - 1 + l \leqslant 2^t + 2l - 1 + 2^t - 1 = \\ = 2^{t+1} + 2l - 2 < 2m - 1.$$

Таким образом, указанная оценка верна. Лемма доказана.

Далее, во время очередного шага алгоритма число единиц в любом векторе p_k, лежащем в рассматриваемой вершине, будет как минимум на один больше, чем число единиц в векторе $p_{left,\,k}$, лежащем в левом потомке данной вершины. Таким образом, в худшем случае после 2^{n-1} шагов алгоритма в левом потомке останутся только векторы с одной единицей. Следовательно, высота дерева ограничивается числом $\min\{2^{n-1},\ 2m - 1\}$. Обозначим его H.

Правая ветвь дерева будет короче, а именно, ее высота будет не больше, чем $\lfloor \log_2 m \rfloor$, так как на каждом шаге алгоритма векторов становится в 2 раза меньше.

Следовательно, рассматриваемое бинарное дерево имеет не больше, чем $2^H - 1$ узлов. Так как в листовых узлах никакие операции не проводятся, число узлов, в которых выполняются операции, можно уменьшить на $2^{H-1} : 2^H - 1 - 2^{H-1} = 2^{H-1} - 1$.

Получается, всего во время выполнения алгоритма будет проведено не больше, чем $m\left(m + \frac{1}{2}\right)2^{H+n-1}$ операций. Данный результат можно переписать как $O\left(m^2 2^{H+n-1}\right)$. Утверждение доказано.

После построения бинарного дерева по указанному выше алгоритму логическая схема синтезируется следующим образом: строится дешифратор, после чего для корня дерева вызывается рекурсивная процедура, описанная в алгоритме 2.

Алгоритм 2. Синтез схемы по бинарному дереву

```
1:  procedure Synthesize (node)
2:  Для каждой пары векторов p_i, p_j
3:  if left = 0 и right = 0 then ◁ этот узел — лист
4:    Синтезировать логическую схему LC_i для p_i
5:    Синтезировать логическую схему LC_j для p_j
6:  return
7:  else
8:    Synthesize (right)
9:    Synthesize (left)
10:   Синтезировать схему LC_i = LC_left_i ∨ LC_right_k
11:   Синтезировать схему LC_j = LC_left_j ∨ LC_right_k
```

После этого следует проверить, существует ли такое k, $k = 1, ..., 2^n$, что $p_{1_k} = ... = p_{m_k} = 0$. Если такое k существует, то из дешифратора можно убрать вычисление соответствующего конъюнкта.

Доказательство корректности алгоритмов 1 и 2

Утверждение 3. Схема, полученная в результате работы алгоритма 2, примененного к результату алгоритма 1, реализует все m требуемых функций.

Доказательство. Вместо функций будем рассматривать их векторы значений. Докажем по индукции, что для векторов $p_1, ..., p_m$, лежащих в корне, схема, синтезированная по алгоритму 2, реализует все функции, которым соответствуют данные векторы значений.

База индукции. Для листовых вершин схемы, соответствующие множеству векторов, лежащих в каждом из листьев дерева, построенного по алгоритму 1, представляют собой простые схемы, реализующие СДНФ, соответствующие данным векторам длины 2^n. Сведения о том, как устроены такие схемы, подробно рассказаны в начале статьи.

Индуктивный переход. Пусть утверждение верно для всех схем, построенных на n-м шаге алгоритма 2, для всех узлов, удаленных от листьев на n ярусов. Докажем, что схема, полученная на $(n + 1)$-м шаге алгоритма 2, реализует все функции, которым соответствуют векторы значений, лежащие в вершине дерева, удаленной от листьев на $n + 1$ ярус. Рассмотрим произвольный вектор p_k^{n+1}, $1 \leqslant k \leqslant m$.

По построению схемы, для того чтобы получить нужный вектор p_k^{n+1}, нужно рассмотреть правого и левого потомков вершины, в которой лежит данный вектор. Затем нужно взять дизъюнкцию двух векторов с номерами k и $\lceil k/2 \rceil$ для левого и правого потомков соответственно.

Опишем более подробно: пусть во время работы алгоритма 1 в рассматриваемой вершине пару с вектором p_k^{n+1} образовал вектор p_r^{n+1}. Это означает, что вектор, находящийся в левом потомке под номером k, получился по формуле

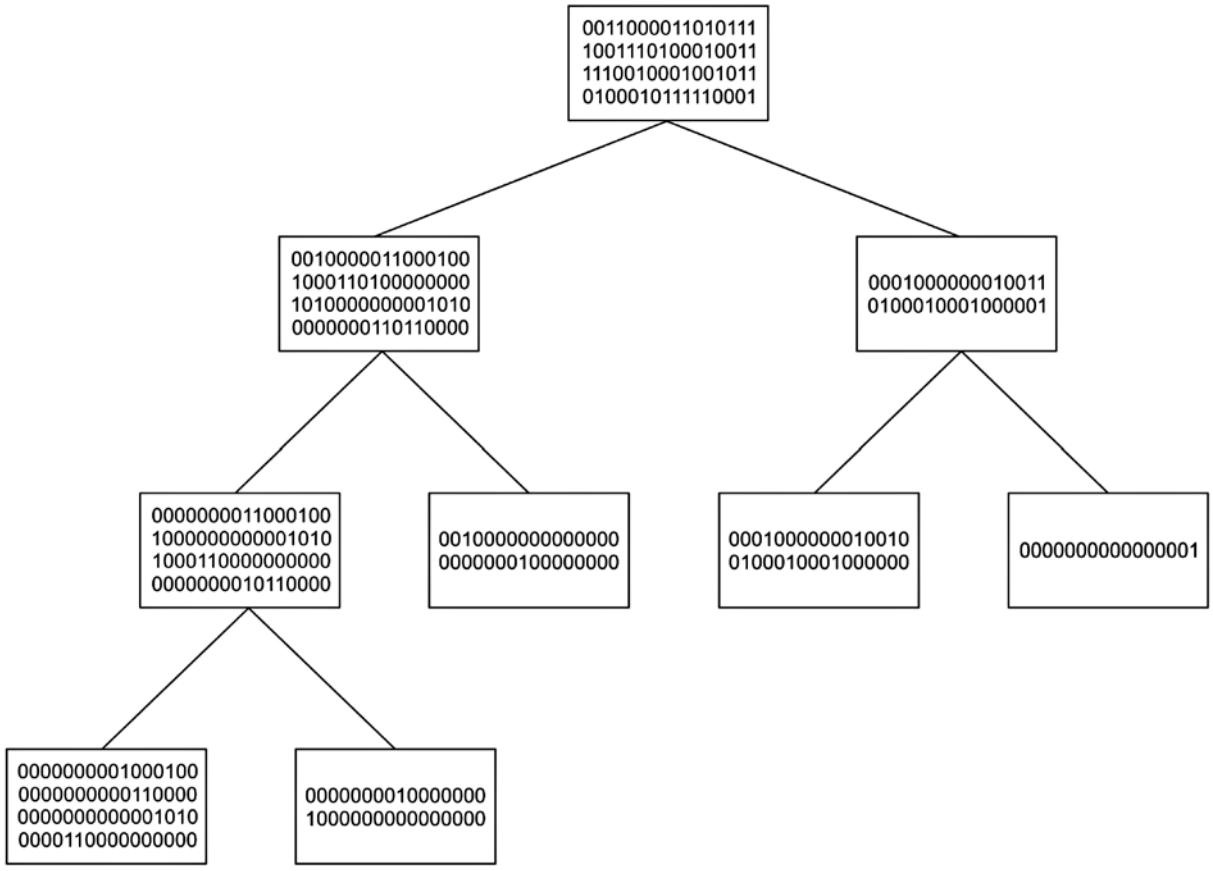

Рис. 4. Результат работы алгоритма 1

$$p^n_{k,\,left} = p^{n+1}_k \,\&\, \overline{p^{n+1}_k \,\&\, p^{n+1}_r} =$$

$$= p^{n+1}_k \,\&\, \left(\overline{p^{n+1}_k} \vee \overline{p^{n+1}_r}\right) = p^{n+1}_k \,\&\, \overline{p^{n+1}_r}.$$

Вектор, находящийся в правом потомке под номером $\lceil k/2 \rceil$, получается при этом по формуле $p^n_{k/2,\,right} = p^{n+1}_k \,\&\, p^{n+1}_r$.

Как следствие, дизъюнкция двух данных векторов равна

$$p^n_{k,left} \vee p^n_{\frac{k}{2},right} = p^{n+1}_k \,\&\, \overline{p^{n+1}_r} \vee p^{n+1}_k \,\&\, p^{n+1}_r =$$

$$= p^{n+1}_k \,\&\, \left(p^{n+1}_r \vee \overline{p^{n+1}_r}\right) = p^{n+1}_k.$$

Следовательно, схема, построенная на $(n + 1)$-м шаге алгоритма 2, реализует функцию с вектором значений p^{n+1}_k. Утверждение доказано.

Пример работы алгоритма

Для демонстрации работы алгоритма рассмотрим S-блок S_1 размером 4×4 бита, который используется в алгоритме Магма [8]. Данное преобразование задается подстановкой {6, 12, 5, 3, 2, 14, 0, 10, 9, 13, 8, 11, 4, 1, 7, 15}. Векторы p_1, p_2, p_3, p_4 для функций выходных битов имеют следующий вид:

$$p_1 = 0011000011010111,$$

$$p_2 = 1001110100010011,$$

$$p_3 = 1110010001001011,$$

$$p_4 = 0100010111110001.$$

Дерево, полученное в результате применения алгоритма 1 к данной системе векторов, изображено на рис. 4.

Глубина схемы, синтезированной по алгоритму 2, равна 6, а сложность — 45. При этом глубина дешифратора равна 2, а сложность — 23 (один конъюнкт можно не вычислять, так как $p_{1_j} = p_{2_j} = p_{3_j} = p_{4_j} = 0$ для $j = 7$).

Применение алгоритма к S-блокам

Сначала рассмотрим несколько полезных свойств S-блоков, которые вытекают из утверждения 1.

Следствие 1. Для любого фиксированного $n > 2$ для каждого S-блока размером $n \times n$ бит бинарные деревья, получаемые в результате работы алгоритма 1, изоморфны.

Доказательство. Для каждого из выходных битов S-блока с номером i, $i = 1, ..., n$ рассмотрим вектор значений p_i длины 2^n. Запишем эти n векторов в виде матрицы:

$$P = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,2^n} \\ \vdots & \ddots & \vdots \\ p_{n,1} & \cdots & p_{n,2^n} \end{pmatrix}.$$

Нетрудно убедиться в том, что все рассматриваемые S-блоки отличаются только порядком столбцов в матрице P. Следовательно, число единиц и нулей в строках данной матрицы будет одинаковым для любого S-блока, а также для любой пары векторов сохраняется число общих единиц. Из этого факта следует, что результат первого шага алгоритма 1 будет одинаков для любого рассматриваемого S-блока размером $n \times n$ бит.

Докажем теперь по индукции, что любая матрица P_l, получаемая на l-м шаге алгоритма, для любого S-блока будет отличаться только перестановкой столбцов. Из этого будет следовать, что алгоритм работает одинаково для любого S-блока на каждом шаге.

База индукции. Для корня требуемое утверждение уже доказано.

Индуктивный переход. Рассмотрим матрицу P_k, которая содержит все векторы из вершины дерева с номером k, и которая была получена на l-м шаге алгоритма. Если данная вершина листовая, ничего доказывать не нужно. В противном случае после очередного шага алгоритма у данной вершины будет два потомка с номерами — $2k$ для левого и $2k + 1$ для правого. Пусть из матрицы P_k пару образовали векторы с номерами i, j. Зафиксируем номер столбца r. В этом случае для элементов матрицы P_{2k} $p_{i,r}^{2k}$ и $p_{j,r}^{2k}$ и для элемента $p_{i,r}^{2k+1}$ матрицы P_{2k+1} верны следующие формулы:

$$p_{i,r}^{2k} = p_i^k \,\&\, \overline{p_i^k \,\&\, p_j^k} = p_i^k \,\&\, \overline{p_j^k};$$

$$p_{j,r}^{2k} = p_j^k \,\&\, \overline{p_i^k \,\&\, p_j^k} = p_j^k \,\&\, \overline{p_i^k};$$

$$p_{i,r}^{2k+1} = p_i^k \,\&\, p_j^k.$$

Следовательно, r-й столбец матриц P_{2k}, P_{2k+1}, полученных на $(l + 1)$-м шаге алгоритма, зависит только от r-го столбца матрицы P_k. Поэтому для любого S-блока матрицы P_{2k}, P_{2k+1} отличаются только перестановкой столбцов. Значит, алгоритм будет работать одинаково вне зависимости от выбора S-блока. Следствие доказано.

Следствие 2. Для каждого S-блока размером $n \times n$ бит алгоритм 1 работает со сложностью $O\left(m^2 2^{3n-2}\right)$.

Доказательство. Воспользуемся результатом утверждения 2. В данном случае $H = 2n - 1$. Поэтому указанная оценка верна. Следствие доказано.

Оценка глубины и сложности полученных схем

Далее глубину схемы S будем обозначать $D(S)$, а сложность — $L(S)$.

Утверждение 4. Для дешифратора от n переменных $(n \in \mathbb{N})$ справедливы следующие оценки:

- $D\left(D_n\right) \leqslant \lceil \log_2 n \rceil$;
- $L\left(D_n\right) \leqslant 2^{n-1} + 2^n$.

Доказательство. Докажем оценку для глубины дешифратора по индукции.

База. Глубина схемы, изображенной на рис. 1, равна нулю, $D\left(D_1\right) = 0 \leqslant \log_2 1 = 0$; глубина схемы, изображенной на рис. 2, равна 1, $D\left(D_2\right) = 1 \leqslant \log_2 2 = 1$.

Индуктивный переход. Пусть оценка верна для всех $i < n$. Для схемы, изображенной на рис. 3, глубина оценивается следующим образом:

$$D\left(D_n\right) = D\left(D_{\left\lceil \frac{n}{2} \right\rceil}\right) + 1 \leqslant \left\lceil \log_2 \left(\left\lceil \frac{n}{2} \right\rceil \right) \right\rceil + 1 =$$

$$= \left\lceil \log_2 \left(\left\lceil \frac{n}{2} \right\rceil \right) + 1 \right\rceil = \left\lceil \log_2 \left(\left\lceil \frac{n}{2} \right\rceil \right) + \log_2 2 \right\rceil = \left\lceil \log_2 \left(2 \left\lceil \frac{n}{2} \right\rceil \right) \right\rceil.$$

Рассмотрим отдельно случай четного и нечетного n.

$\underline{n = 2k}$:

$$\left\lceil \log_2 \left(2 \left\lceil \frac{n}{2} \right\rceil \right) \right\rceil = \lceil \log_2 (2k) \rceil = \lceil \log_2 (n) \rceil.$$

$\underline{n = 2k + 1}$:

$$\left\lceil \log_2 \left(2 \left\lceil \frac{n}{2} \right\rceil \right) \right\rceil = \lceil \log_2 (2(k+1)) \rceil =$$

$$= \lceil \log_2 (2k+2) \rceil \geqslant \lceil \log_2 (2k+1) \rceil = \lceil \log_2 (n) \rceil.$$

Докажем, что для любого $k \in \mathbb{N}$ $\lceil \log_2 (2k+2) \rceil = \lceil \log_2 (2k+1) \rceil$. Легко заметить, что $\log_2 (2k+1)$ не может быть целым числом. Следовательно, $\lceil \log_2 (2k+1) \rceil = \log_2 (2k+1+t)$ для некоторого натурального t. Вместе с тем $\log_2 (2k+1+t) = \log_2 \left(2k+2+(t-1)\right)$. Значит, при любом натуральном t $\lceil \log_2 (2k+2) \rceil = \lceil \log_2 (2k+1) \rceil$.

Следовательно, оценка глубины верна для $i = n$.

Теперь докажем оценку сложности предложенной схемы. Снова сделаем это по индукции.

База. Сложность схемы, изображенной на рис. 1, равна нулю, $L\left(D_1\right) = 0 \leqslant 2^1 + 2^0 = 3$; сложность схемы, изображенной на рис. 2, равна 4, $L\left(D_2\right) = 4 \leqslant 2^2 + 2^1 = 6$.

Индуктивный переход. Для схемы, изображенной на рис. 3, сложность оценивается следующим образом:

$$L\left(D_n\right) = L\left(D_{\left\lceil \frac{n}{2} \right\rceil}\right) + L\left(D_{\left\lfloor \frac{n}{2} \right\rfloor}\right) + 2^n \leqslant$$

$$\leqslant 2^{\left\lceil \frac{n}{2} \right\rceil} + 2^{\left\lceil \frac{n}{2} \right\rceil - 1} + 2^{\left\lfloor \frac{n}{2} \right\rfloor} + 2^{\left\lfloor \frac{n}{2} \right\rfloor - 1} + 2^n.$$

Так как схемы для D_1 и D_2 строятся нерекурсивно, нужно также отдельно рассмотреть и случаи, когда D_1 или D_2 являются составными блоками схемы на рис. 3 $\left(D_{\left\lceil \frac{n}{2} \right\rceil}$ или $D_{\left\lfloor \frac{n}{2} \right\rfloor}\right)$, т. е. когда $n = 3, 4, 5$:

- $n = 3$, $L\left(D_3\right) = L\left(D_2\right) + L\left(D_1\right) + 2^3 = 4 + 0 + 8 = 12 \leqslant 2^3 + 2^2 = 12$;
- $n = 4$, $L\left(D_4\right) = L\left(D_2\right) + L\left(D_2\right) + 2^4 = 4 + 4 + 16 = 24 \leqslant 2^4 + 2^3 = 24$;

• $n = 5$, $L(D_5) = L(D_3) + L(D_2) + 2^5 \leqslant 12 + 4 + 32 = 48 \leqslant 2^5 + 2^4 = 48$.

Для остальных n рассмотрим отдельно четный и нечетный случаи.

$\underline{n = 2k}$:

$$L(D_n) = 2L(D_k) + 2^{2k} \leqslant 2(2^{k-1} + 2^k) + 2^{2k} = 2^k + 2^{k+1} + 2^{2k} = 3 \cdot 2^k + 2^{2k}.$$

Для того чтобы утверждение было верно, необходимо, чтобы выполнялось неравенство

$$3 \cdot 2^k + 2^{2k} \leqslant 2^{2k-1} + 2^{2k} \Leftrightarrow 3 \cdot 2^k \leqslant 2^{2k-1} \Leftrightarrow$$
$$\Leftrightarrow k + \log_2 3 \leqslant 2k - 1 \Leftrightarrow k \geqslant 1 + \log_2 3 \Leftrightarrow k \geqslant 3.$$

Следовательно, оценка сложности верна для всех четных $n \geqslant 6$.

$\underline{n = 2k + 1}$:

$$L(D_n) = L(D_{k+1}) + L(D_k) + 2^{2k+1} \leqslant$$
$$\leqslant 2^{k+1} + 2^k + 2^k + 2^{k-1} + 2^{2k+1} = 9 \cdot 2^{k-1} + 2^{2k+1}.$$

Для того чтобы утверждение было верно, необходимо, чтобы выполнялось неравенство

$$9 \cdot 2^{k-1} + 2^{2k+1} \leqslant 2^{2k} + 2^{2k+1} \Leftrightarrow 9 \cdot 2^{k-1} \leqslant 2^{2k} \Leftrightarrow$$
$$\Leftrightarrow k - 1 + \log_2 9 \leqslant 2k \Leftrightarrow k \geqslant \log_2 9 - 1 \Leftrightarrow k \geqslant 3.$$

Следовательно, оценка сложности верна для всех нечетных $n \geqslant 7$. Утверждение доказано.

Утверждение 5. Для схемы Φ_n, реализующей все булевы функции от n переменных $(n \in \mathbb{N})$, справедливы следующие оценки:

- $D(\Phi_n) \leqslant n + \lceil \log_2 n \rceil - 1$;
- $L(\Phi_n) \leqslant 2^{2^n - 1} + 2^{n-1} - n - 1$.

Доказательство. Очевидно, для глубины данной схемы верна оценка $D(\Phi_n) \leqslant D(D_n) + n - 1$, так как первый слой входит в дешифратор, а помимо него остается $n - 1$ слой. На каждом из них параллельно вычисляется не более одной дизъюнкции. Следовательно, $D(\Phi_n) \leqslant n - 1 + \lceil \log_2 n \rceil$.

Теперь оценим сложность данной схемы. Обозначим схему, реализующую слой под номером i, R_i, $2 \leqslant i \leqslant n$. Тогда

$$L(\Phi_n) = L(D_n) + \sum_{i=2}^{n} L(R_i).$$

Сложность схемы R_2 не превосходит числа пар из 2^n различных конъюнкций K_i, т. е. $L(R_2) \leqslant C_{2^n}^2$. Аналогично получаем:

- $L(R_3) \leqslant C_{2^n}^3 + C_{2^n}^4$;
- $L(R_d) \leqslant C_{2^n}^{2^{d-2}+1} + \ldots + C_{2^n}^{2^{d-1}-1} + C_{2^n}^{2^{d-1}}$;
- $L(R_n) \leqslant C_{2^n}^{2^{n-2}+1} + \ldots + C_{2^n}^{2^{n-1}-1} + \frac{1}{2} C_{2^n}^{2^{n-1}} - n$.

В итоге получаем (сложность можно уменьшить еще на n, так как селекторы вычислять не нужно):

$$\sum_{i=2}^{n} L(R_i) \leqslant \sum_{i=2}^{2^{n-1}-1} C_{2^n}^i + \frac{1}{2} C_{2^n}^{2^{n-1}} - n.$$

Так как $C_n^k = C_n^{n-k}$ при $0 \leqslant k \leqslant n$ и $\sum_{i=0}^{n} C_n^i = 2^n$, легко заметить, что

$$\sum_{i=2}^{n} L(R_i) \leqslant \sum_{i=2}^{2^{n-1}-1} C_{2^n}^i + \frac{1}{2} C_{2^n}^{2^{n-1}} - n =$$
$$= \frac{1}{2} \cdot 2^{2^n} - 2^n - n - 1 = 2^{2^n - 1} - 2^n - n - 1.$$

Следовательно,

$$L(\Phi_n) \leqslant 2^{n-1} + 2^n + 2^{2^n-1} - 2^n - n - 1 =$$
$$= 2^{2^n-1} + 2^{n-1} - n - 1.$$

Утверждение доказано.

Следствие 3. Для схемы AF_n, реализующей все дизъюнкции n элементов $(n \in \mathbb{N})$, справедливы следующие оценки:

- $D(AF_n) \leqslant n$,
- $L(AF_n) \leqslant 2^n - n - 1$.

Утверждение 6. Для схемы произвольного сбалансированного S-блока от n переменных $(n \in \mathbb{N})$, полученной в результате работы программы с использованием наивного метода синтеза, верны следующие оценки:

- $D \leqslant \lceil \log_2 n \rceil + n - 1$;
- $L \leqslant 2^n + (n+1)(2^{n-1} - 1) + 1$.

Доказательство. Из утверждения 1 следует, что схему дешифратора можно построить с глубиной $\lceil \log_2 n \rceil$ и сложностью $2^n + 2^{n-1}$. Вычисления на шаге 3 можно сделать на глубине $\log_2 2^{n-1} = n - 1$ и со сложностью $n(2^{n-1} - 1)$.

Следовательно, указанные оценки верны. Утверждение доказано.

Утверждение 7. Для схемы произвольного сбалансированного S-блока от n переменных $(n \in \mathbb{N})$, полученной в результате работы программы с использованием улучшенного наивного метода синтеза, верны следующие оценки:

- $D \leqslant \lceil \log_2 n \rceil + n - 1$;
- $L \leqslant 2^n + (n+1)2^{n-1} - (n-1)(2^{n-3} + 1)$.

Доказательство. Для глубины оценка, очевидно, остается такой же, как и для наивного метода.

Для оценки сложности рассмотрим отдельно случаи четного и нечетного n. Очевидно, что и в случае $n = 2k$, и в случае $n = 2k + 1$ число общих пар равно k. Каждая общая пара уменьшает сложность на 2^{n-2}.

Используя это и утверждение 5, получаем следующие оценки.

$\underline{n = 2k}$:

$$L \leqslant 2^{2k} + (2k+1)\left(2^{2k-1} - 1\right) + 1 - k2^{2k-2} =$$
$$= 2^{2k} + (2k+1)\left(2^{2k-1} - 1\right) + 1 - 2k2^{2k-3} =$$
$$= 2^{2k} + 2k2^{2k-1} + 2^{2k-1} - 2k - 1 + 1 - 2k2^{2k-3} =$$
$$= 2^{2k} + (2k+1)2^{2k-1} - 2k\left(2^{2k-3} + 1\right) \leqslant 2^{2k} +$$
$$+ (2k+1)2^{2k-1} - 2k2^{2k-3} = 2^n + (n+1)2^{n-1} - n2^{n-3}.$$

$\underline{n = 2k+1}$:

$$L \leqslant 2^{2k+1} + (2k+2)\left(2^{2k} - 1\right) + 1 - k2^{2k-1} =$$
$$= 2^{2k+1} + (2k+2)\left(2^{2k} - 1\right) + 1 - 2k2^{2k-2} =$$
$$= 2^{2k+1} + (2k+2)2^{2k} - 2k - 2 + 1 - 2k2^{2k-2} =$$
$$= 2^{2k+1} + (2k+1)2^{2k-1} - 2k\left(2^{2k-2} + 1\right) - 1 \leqslant$$
$$\leqslant 2^{2k+1} + (2k+2)2^{2k} - 2k2^{2k-2} =$$
$$= 2^n + (n+1)2^{n-1} - (n-1)2^{n-3}.$$

Таким образом, указанная оценка верна. Утверждение доказано.

Утверждение 8. Для схемы произвольного сбалансированного S-блока от n переменных $(n \in \mathbb{N})$, полученной в результате работы программы с использованием метода синтеза Шеннона, верны следующие оценки:

- $D \leqslant \min\left(k + \lceil \log_2 k \rceil - 1, \lceil \log_2 (n-k) \rceil\right) + 1 + n - k;$
- $L \leqslant 2^{2^k - 1} + 2^{k-1} + (2n+1)2^{n-k} + 2^{n-k-1} - n - k - 1.$

Доказательство. Поэтапно оценим глубину и сложность реализации произвольного S-блока данным методом.

- Из утверждений 1 и 2 следует, что $D(\Phi_k) \leqslant k + \lceil \log_2 k \rceil - 1$, $L(\Phi_k) \leqslant 2^{2^k - 1} + 2^{k-1} - k - 1$, $D(D_{n-k}) \leqslant \lceil \log_2 (n-k) \rceil$, $L(D_{n-k}) \leqslant 2^{n-k} + 2^{n-k-1}$.
- 2^{n-k} конъюнкций с необходимой функцией $f(x_1, \ldots, x_k)$ выполняются на глубине 1 со сложностью 2^{n-k} (данные вычисления необходимо провести для каждого из n выходов S-блока).
- Вычисление дизъюнкции 2^{n-k} элементов, полученных на прошлом шаге, выполняется на глубине $\lceil \log_2 2^{n-k} \rceil = n - k$ и со сложностью $2^{n-k} - 1$ (данные вычисления также надо повторить для каждого выхода S-блока).

Так как Φ_k и D_{n-k}, а также значение каждого выхода S-блока вычисляются параллельно, то общая глубина его реализации оценивается следующим образом:

$$D(S_n) \leqslant \min\left(k + \lceil \log_2 k \rceil - 1, \lceil \log_2 (n-k) \rceil\right) + 1 + n - k.$$

Для сложности верна следующая оценка:

$$L(S_n) \leqslant 2^{2^k - 1} + 2^{k-1} - k - 1 + 2^{n-k} +$$
$$+ 2^{n-k-1} + n\left(2^{n-k} + 2^{n-k} - 1\right) =$$
$$= 2^{2^k - 1} + 2^{k-1} + (2n+1)2^{n-k} + 2^{n-k-1} - n - k - 1.$$

Утверждение доказано.

Утверждение 9. Для схемы произвольного сбалансированного S-блока от n переменных $(n \in \mathbb{N})$, полученной в результате работы программы с использованием метода синтеза Лупанова, верны следующие оценки:

- $D \leqslant \min\left(\begin{array}{l} \lceil \log_2 k \rceil + \lceil \log_2 s \rceil + \\ + \left\lceil \log_2 \left\lceil \frac{2^k}{s} \right\rceil \right\rceil, \lceil \log_2 (n-k) \rceil \end{array}\right) + 1 + n - k;$
- $L \leqslant n\left\lceil \frac{2^k}{s} \right\rceil \min\left(2^s - s - 1,\ n(s-1)2^{n-k}\right) +$
$$+ \left(n\left(\left\lceil \frac{2^k}{s} \right\rceil + 1\right) + 1\right)2^{n-k} + 2^{n-k-1} + 2^k + 2^{k-1} - n.$$

Доказательство. Поэтапно оценим глубину и сложность реализации произвольного S-блока данным методом:

Шаг 1. Из утверждения 4 следует, что $D(D_k) \leqslant \lceil \log_2 k \rceil$, $L(D_k) \leqslant 2^k + 2^{k-1}$.

Шаг 2. Выполняя те же действия, что и при доказательстве утверждения 5, для некоторой полосы можно получить все возможные функции $f_{ij}(x_1, \ldots, x_k)$. Единственное отличие состоит в том, что добавится еще один слой вычислений, так как оставшуюся половину функций нельзя получить с помощью отрицания в силу того, что оно затронет не только значения рассматриваемой полосы, но и все остальные. Следовательно, вычислить все возможные функции $f_{ij}(x_1, \ldots, x_k)$ для одной полосы можно на глубине $\lceil \log_2 s \rceil$ и со сложностью $2^s - s - 1$ (так как 0 и все функции, равные 1 на одном наборе, уже реализованы в дешифраторе). Однако при большом s и k, близком к n, может оказаться, что проще посчитать по СДНФ функции для каждого столбца (всего будет $n2^{n-k}$ столбцов, 2^{n-k} для каждого выхода S-блока). Это можно сделать на глубине $\lceil \log_2 s \rceil$ и со сложностью $n(s-1)2^{n-k}$. Для каждой полосы эти вычисления можно проводить параллельно, следовательно, данный этап можно выполнить на глубине $\lceil \log_2 s \rceil$ и со сложностью $p \min(2^s - s - 1,\ n(s-1)2^{n-k})$.

Шаг 3. Для каждого столбца функция $f(x_1, \ldots, x_k, \sigma_{k+1}, \ldots, \sigma_n)$ получается с помощью $(p-1)$ дизъюнкции функций, полученных на предыдущем шаге. Следовательно, данный этап можно реализовать на глубине $\lceil \log_2 p \rceil$ и со сложностью $(p-1)2^{n-k}$ (эти вычисления надо повторить для каждого выхода S-блока).

Шаг 4. Из утверждения 4 следует, что $D(D_{n-k}) \leqslant \lceil \log_2 (n-k) \rceil$, $L(D_{n-k}) \leqslant 2^{n-k} + 2^{n-k-1}$.

Шаг 5. 2^{n-k} конъюнкций с необходимой функцией $f(x_1, \ldots, x_k)$ выполняются на глубине 1 со сложностью 2^{n-k} (данные вычисления необходимо провести для каждого из n выходов S-блока).

Шаг 6. Вычисление дизъюнкции 2^{n-k} элементов, полученных на прошлом шаге, выполняется на глубине $\lceil \log_2 2^{n-k} \rceil = n - k$ и со сложностью $2^{n-k} - 1$ (данные вычисления также надо повторить для каждого выхода S-блока).

Так как вычисления на шаге 1—3 и на шаге 4 проводятся параллельно, а также значение каждого выхода S-блока вычисляются параллельно, то общая глубина его реализации оценивается следующим образом:

$$D(S_n) \leqslant \min\left(\begin{array}{l}\lceil \log_2 k \rceil + \lceil \log_2 s \rceil + \\ + \left\lceil \log_2 \left\lceil \frac{2^k}{s} \right\rceil \right\rceil, \lceil \log_2 (n-k) \rceil\end{array}\right) + 1 + n - k.$$

Для сложности верна следующая оценка:

$$L(S_n) \leqslant 2^k + 2^{k-1} + np \min\left(2^s - s - 1,\ n(s-1)2^{n-k}\right) +$$
$$+ n(p-1)2^{n-k} + 2^{n-k} + 2^{n-k-1} +$$
$$+ n\left(2^{n-k} + 2^{n-k} - 1\right) = n\left\lceil \frac{2^k}{s} \right\rceil \min\left(2^s - s - 1,\ n(s-1)2^{n-k}\right) +$$
$$+ \left(n\left(\left\lceil \frac{2^k}{s} \right\rceil + 1\right) + 1\right)2^{n-k} + 2^{n-k-1} + 2^k + 2^{k-1} - n.$$

Утверждение доказано.

Сравнение практической эффективности алгоритмов

Для того чтобы понять, насколько эффективны приведенные выше алгоритмы применительно к S-блокам, рассмотрим следующие подстановки:

1) S_{GOST} — S-блок размером 8×8 бит, используемый в блочном шифре Кузнечик [8] и хеш-функции Стрибог [9];

2) S_{AES} — S-блок размером 8×8 бит, используемый в блочном шифре AES [10];

3) S_{SHOW} — S-блок размером 8×8 бит, используемый в потоковом шифре SNOW3G (в спецификации обозначен S_Q) [11];

4) S_{ZUC_1} — S-блок размером 8×8 бит, используемый в потоковом шифре ZUC (в спецификации обозначен S_0) [12];

5) S_{ZUC_2} — S-блок размером 8×8 бит, используемый в потоковом шифре ZUC (в спецификации обозначен S_1) [12];

6) S_{KASUMI_1} — S-блок размером 7×7 бит, используемый в потоковом шифре KASUMI [13];

7) S_{KASUMI_2} — S-блок размером 9×9 бит, используемый в потоковом шифре KASUMI [13].

Для каждого алгоритма посчитаем глубину и сложность схемной реализации всех S-блоков, перечисленных выше. В дополнение к изложенным выше алгоритмам приведем результат построения схемы с помощью программы Logic Friday [14] с использованием совместной минимизации ДНФ всех выходов S-блока.

Для метода Шеннона во всех случаях лучший результат был показан при $k = 3$. Для метода Лупанова лучшими в плане сложности были следующие параметры:

- $n = 7$: $k = 4$, $s = 4$;
- $n = 8$: $k = 4$, $s = 6$;
- $n = 9$: $k = 4$, $s = 6$.

Рассмотрим отдельно сложность и глубину.

Как видно из данных табл. 1, во все случаях метод нахождения ДНФ, близкой к минимальной, для каждого выхода S-блока по отдельности позволяет добиться минимальной глубины реализации.

Из данных табл. 2 следует, что лучший результат в плане сложности во всех случаях показывает метод Лупанова. Однако метод Шеннона совсем немного проигрывает ему в плане сложности, однако он лучше в плане глубины.

Авторский алгоритм показывает сложность, сопоставимую с Logic Friday, и лучшую глубину. При этом данный алгоритм позволяет получить реализацию с гарантированной сложностью для любого S-блока при фиксированном n.

Таблица 1

Глубина реализации S-блоков

S-блок	Наивный метод (+улучшенный)	Метод упрощения ДНФ	Logic Friday	Метод Шеннона	Метод Лупанова	Авторский алгоритм
S_{GOST}	10	9	16	10	12	11
S_{AES}	10	9	16	10	12	11
S_{SNOW}	10	9	16	10	12	11
S_{ZUC_1}	10	9	17	10	12	11
S_{ZUC_2}	10	9	18	10	12	11
S_{KASUMI_1}	9	8	15	9	10	10
S_{KASUMI_2}	11	10	—	11	13	13
Прочерк означает, что программа Logic Friday "виснет" при работе с S-блоком размером 9×9 бит.						

Таблица 2

Сложность реализации S-блоков

S-блок	Наивный метод	Улучшенный наивный метод	Метод упрощения ДНФ	Logic Friday	Метод Шеннона	Метод Лупанова	Авторский алгоритм
S_{GOST}	1319	1068	973	823	680	677	838
S_{AES}	1319	1068	973	780	680	677	838
S_{SNOW}	1319	1068	960	800	680	677	838
S_{ZUC_1}	1319	1068	929	747	680	677	838
S_{ZUC_2}	1319	1068	987	862	680	677	838
S_{KASUMI_1}	604	512	463	461	372	353	416
S_{KASUMI_2}	2878	2371	1846	—	1359	1340	1750

Таблица 3

Время работы алгоритма на основе метода упрощения ДНФ

Размер S-блока, *n*	Время работы алгоритма на основе упрощения ДНФ, с
16	9866
15	2295
14	474
13	122
12	29
11	7
4...10	< 5

Для измерения производительности программы было сгенерировано по десять случайных S-блоков размером от 4×4 до 16×16 бит. Тестирование проводилось на ПК со следующими характеристиками:

- процессор Intel(R) Core(TM) i7-2600K CPU@ 3.40GHz;
- один поток;
- оперативная память 16GB 1067 MHz DDR3.

При построении схем для S-блоков размером от 4×4 до 16×16 бит любым методом, кроме упрощения ДНФ, программа работает менее 5 с. Результаты для данного метода указаны в табл. 3.

Заключение

Представлены результаты исследований различных методов аппаратной реализации сбалансированных S-блоков. В качестве параметров, характеризующих эффективность схемы, используются ее глубина и сложность.

Рассмотрены стандартные методы синтеза (на основе СДНФ, Шеннона, Лупанова), предложен новый алгоритм для произвольной системы булевых функций и способ уменьшить сложность схемы, получаемой после упрощения алгоритмом ESPRESSO ДНФ функции, соответствующей каждому выходу S-блока. Для сравнения эффективности методов написана программа, генерирующая схему на языке Verilog. Для схем, генерируемых в результате работы программы, получены оценки глубины и сложности, доказана их корректность.

Автор выражает благодарность своему научному руководителю, канд. физ.-мат. наук, доц. А. В. Галатенко за внимание к работе и замечания, высказанные в ходе подготовки статьи.

Список литературы

1. **Болотов А. А., Галатенко А. В., Гринчук М. И., Золотых А. А., Иванович Л.** Методы оптимизации глубины реализации хэш-функций // Интеллектуальные системы. — 2013. — Т. 17, № 1—4. — С. 224—228.

2. **Курганов Е. А.** О глубине аппаратной реализации блочного шифра Кузнечик // Интеллектуальные системы. — 2016. — Т. 20, № 1. — С. 61—76.

3. **Супрунюк С. О., Курганов Е. А.** О глубине аппаратной реализации потокового шифра ZUC // Программная инженерия. — 2018. — Т. 9, № 5. — С. 221—227.

4. **Huffman D.** Canonical Forms for Information-Lossless Finite-State Logical Machines. // IRE Transactions on Circuit Theory. — 1959. — Vol. 6, No. 5. — P. 41—59.

5. **Rudell R. L., Sangiovanni-Vincentelli A.** Multiple-valued minimization for PLA optimization // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. — 1987. — Vol. 6, No. 5. — P. 727—750.

6. **Яблонский С. В.** Введение в дискретную математику. — М.: Наука, 1986. 384 с.

7. **Borisenko N. P., Tho H. D.** Algorithm for minimization of the number of logic elements in a circuit implementing S-box Boolean functions // 3rd Workshop on Current Trends in Cryptology (CTCrypt 2014). — 2014. — P. 125—144.

8. **ГОСТ Р 34.12—2015.** Информационная технология. Криптографическая защита информации. Блочные шифры. Государственный стандарт РФ, 2015.

9. **ГОСТ Р 34.11—2012.** Информационная технология. Криптографическая защита информации. Функция хэширования. Государственный стандарт РФ, 2012.

10. **National** Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES). FIPS-197, 2001.

11. **Specification** of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW 3G Specification, version 1.1, 2006.

12. **Specification** of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification, version 1.5, 2011.

13. **KASUMI** specification, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2, ETSI/ SAGE, 1999.

14. **Logic** Friday. URL: https: // download.cnet.com/developer/logic-friday/i-10268041.

On Hardware Implementation of Balanced S-boxes

E. A. Kurganov, kuev@yandex.ru, Scientific and Research Institute Voskhod, Moscow, 119607, Russian Federation

Corresponding author:

Kurganov Evgeny A., Lead Software Developer, Scientific and Research Institute Voskhod, Moscow, 119607, Russian Federation
E-mail: kuev@yandex.ru

Received on September 24, 2020
Accepted on October 07, 2020

An S-box is a non-linear transformation that takes n bits as input and returns m bits. This transformation is most easily represented as a n×m lookup table. Most often, only balanced S-boxes are used in cryptography. This means that the number of input bits is equal to the number of output bits. The S-box is an important part of most symmetric ciphers. The selection of the correct substitution makes the link between the key and the ciphertext more complex (non-linear), which makes it much more difficult to hack.

This paper deals with a hardware implementation of S-boxes. This implementation can be realized by using logical conjunction, disjunction, negation and delay blocks. The main indicator of productivity of such implementations is a circuit depth, namely the maximum length of a simple way of the circuit and a circuit complexity, namely the quantity of logic elements (negation elements are not taken into account).

The article considers the standard synthesis methods (based on DNF, Shannon, Lupanov), proposes a new algorithm to minimize the complexity of an arbitrary Boolean functions system and a way to reduce the complexity of the circuit obtained after simplification by the ESPRESSO algorithm of DNF of the function related to the output of the S-box. To compare the efficiency of the methods, the C++ program was created that generates a circuit in the Verilog language. The estimates of depth and complexity are obtained for the schemes produced as a result of the program's operation. The article ends with a comparison of the efficiency of S-box schemes of known cryptographic standards obtained as the output of the program (with each other and with the result of the Logic Friday program).

Keywords: *S-box, hardware implementation, circuit depth optimization, circuit complexity optimization, stream ciphers, block ciphers*

For citation:

Kurganov E. A. On Hardware Implementation of Balanced S-boxes, *Programmnaya Ingeneria,* 2021, vol. 12, no. 1, pp. 8—20

DOI: 10.17587/prin.12.8-20

References

1. **Bolotov A. A., Galatenko A. V., Grinchuk M. I., Zolotykh A. A., Ivanovich L.** Techniques for depth optimization of hash-functions implementations, *Intellektualnye Sistemy*, 2013, vol. 17, no. 1—4, pp. 224—228 (in Russian).

2. **Kurganov E. A.** On depth of the Kuznyechik block cipher hardware implementation, *Intellektualnye Sistemy*, 2016, vol. 20, no. 1, pp. 61—76 (in Russian).

3. **Suprunyuk S. O., Kurganov E. A.** On depth of the ZUC stream cipher hardware implementation, *Programmnaya Ingeneria*, 2018, vol. 9, no. 5, pp. 221—227 (in Russian).

4. **Huffman D.** Canonical Forms for Information-Lossless Finite-State Logical Machines, *IRE Transactions on Circuit Theory*, 1959, vol. 6, no. 5, pp. 41—59.

5. **Rudell R. L., Sangiovanni-Vincentelli A.** Multiple-valued minimization for PLA optimization, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 1987, vol. 6, no. 5, pp. 727—750.

6. **Jablonskii S. V.** *Introduction to discrete mathematics.* Moscow, Nauka, 1986, 384 p. (in Russian).

7. **Borisenko N. P., Tho H. D.** Algorithm for minimization of the number of logic elements in a circuit implementing S-box Boolean functions, 3*rd Workshop on Current Trends in Cryptology (CTCrypt 2014),* 2014, pp. 125—144.

8. **GOST R 34.12—2015,** "Information technology. Cryptographic data security. Block ciphers", Federal Agency on Technical Regulating and Metrology, 2015.

9. **GOST R 34.11—2012,** "Information technology. Cryptographic data security. Hashing function", Federal Agency on Technical Regulating and Metrology, 2012.

10. **National** Institute of Standards and Technology (NIST). Advanced Encryption Standard (AES). FIPS-197, 2001.

11. **Specification** of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW 3G Specification, version 1.1, 2006.

12. **Specification** of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification, version 1.5, 2011.

13. **KASUMI** specification, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2, ETSI/SAGE, 1999.

14. **Logic** Friday, available at: https: // download.cnet.com/developer/logic-friday/i-10268041.

УДК 532.517 DOI:10.17587/prin.12.21-30

В. Г. Лущик, д-р техн. наук, вед. науч. сотр., vgl_41@mail.ru,
М. С. Макарова, канд. техн. наук, науч. сотр., april27_86@mail.ru,
А. И. Решмин, канд. техн. наук, вед. науч. сотр., alexreshmin@rambler.ru,
Научно-исследовательский институт механики МГУ имени М. В. Ломоносова

Применение трехпараметрической дифференциальной модели турбулентности для решения задач течения и теплообмена в каналах переменного сечения. Часть 1

Представлена методика численного исследования в приближении узкого канала для задач течения и теплообмена в плоских и круглых каналах переменного сечения с использованием дифференциальной трехпараметрической модели сдвиговой турбулентности. Описаны основные результаты многочисленных исследований с использованием предложенной методики, одной из целей проведения которых стало обоснование возможности использования приближения узкого канала. Обзорное исследование выполнено в двух частях. В настоящей части приведены результаты исследований смешанной конфекции в вертикальных трубах в условиях устойчивой и неустойчивой стратификации, а также течений в каналах с проницаемыми стенками при наличии вдува или отсоса на стенке.

Ключевые слова: *дифференциальная модель турбулентности, диффузор, конфузор, градиент давления, вдув, отсос, ламинаризация течения, интенсификация теплообмена*

Введение

Существующие к настоящему времени методы расчета течения и теплообмена в каналах постоянного и переменного сечения в приближении пограничного слоя (ПС) (см., например, работу авторов [1] и список литературы к ней) позволяют определить необходимые для практики интегральные и локальные характеристики сопротивления и теплообмена. Однако очевидным недостатком приближения ПС, использующим разбиение потока на ядро и пограничный слой, является необходимость взаимной корректировки решений для каждой из этих областей. Этот недостаток увеличивается в случае канала переменного сечения и при использовании завесного охлаждения, когда часть расхода поступает в канал через одну или несколько завес, что приводит к существенному изменению параметров ПС и ядра потока, т. е. к задаче не только о канале переменного сечения, но и переменного по длине расхода. Для преодоления этого недостатка приближения ПС могут быть использованы методы сквозного счета от стенки до оси сопла, простейшим из которых является приближение узкого канала (УК), базирующееся на тех же уравнениях, что и приближение ПС.

Для плоских и цилиндрических каналов постоянного сечения использование приближения УК оказалось весьма продуктивным. Проведенные с использованием трехпараметрической модели турбулентности [2] расчеты развития течений в каналах с однородным и неоднородным профилями скорости и характеристик турбулентности на входе, обзор которых сделан в работе [3], позволили впервые численно описать ряд особенностей течений, полученных в эксперименте, в частности, немонотонное изменение скорости и характеристик турбулентности по длине [3]. Обзор результатов по развитию ламинарных течений в каналах содержится в работе [4], где указаны границы применимости приближения УК, определенные путем сравнения с результатами, полученными на основе полной системы уравнений Навье—Стокса.

Что касается использования приближения УК в расчетах течений в каналах переменного сечения, в том числе и в соплах жидкостных ракетных двигателей (ЖРД), то применимость его не является очевидной, по крайней мере, по двум причинам. Во-первых, система уравнений движения и энергии, записанная в цилиндрических (для осесимметричного сопла) координатах в приближении УК, не является эквивалентной аналогичной системе в приближении ПС, записанной в координатах, связанных со стенкой сопла, особенно при больших углах наклона стенки. Во-вторых, распределение давления в сопле, одномерное в случае использования приближения УК, будет отличаться от двумерного, которое получается при расчете невязкого течения в ядре потока и используется при расчетах в приближении ПС. Попытки использования приближения УК для расчета течения в соплах известны [4, 5]. В работе [5] представлены результаты расчетов ламинарного течения в соплах Лаваля и указаны границы применимости приближения УК.

Анализ применимости этого приближения для расчета турбулентного течения в соплах ЖРД приведен в работе [6]. Проведенные в этом исследовании расчеты течения в сопле ЖРД с большим раскрытием (~10) и сравнение по интегральным и локальным характеристикам (толщинам вытеснения и потери импульса, тепловому потоку в стенку, коэффициентам трения, потери импульса в силу трения) с результатами расчета в приближении пограничного слоя с одномерным и двумерным ядром потока позволили установить следующее.

Максимальное различие результатов расчетов (по толщине потери импульса и коэффициенту потери импульса в силу трения) в приближениях узкого канала и ПС с одномерным ядром потока не превышает 10...13 % на выходе из сопла. Этот факт означает, что различие в системах координат, использованных в этих приближениях, невелико и позволяет считать их практически эквивалентными при проведении расчетов. Различие результатов расчета, полученных в приближениях УК и ПС с двумерным ядром потока, получилось большим. Так, по значению теплового потока в районе критического сечения сопла приближение УК дает более высокие (более чем в 2 раза), а на остальной части сопла более низкие (в 1,5 раза) значения теплового потока. По значению коэффициента потери импульса в силу трения получено занижение результатов примерно на 30 % в выходном сечении сопла. Указанные различия относятся как к соплу без завесы, так и с завесой в докритической части сопла. Полученное различие результатов в этих приближениях объясняется существенным отличием чисел Маха и значений давлений по длине сопла при одномерном в приближении УК, и двумерном, использованном в приближении ПС, подходах.

Целью настоящей работы является краткий обзор результатов численного исследования течения и теплообмена, использующего приближение УК для расчета турбулентного течения в каналах постоянного и переменного сечения с различными граничными условиями. Численное исследование выполнено по разработанной авторами методике, использующей трехпараметрическую модель турбулентности [2]. Представлены результаты численного моделирования с использованием предложенной методики для решения задач о смешанной конвекции в вертикальных трубах, влиянии вдува и отсоса на течение в трубе, ламинаризации потока при течении в конфузоре и трубе, интенсификации теплообмена при течении в диффузорах и в пластинчатом теплообменнике с диффузорными каналами.

Результаты исследования представлены в двух публикациях. В части 1 приведено подробное описание самой вычислительной методики и результаты исследований смешанной конфекции в вертикальных трубах в условиях устойчивой и неустойчивой стратификации, а также течений в каналах с проницаемыми стенками при наличии вдува или отсоса на стенке.

1. Методика расчета

С использованием трехпараметрической дифференциальной модели турбулентности [2] в работе [1] в приближении ПС описана разработанная авторами методика численного исследования задач внешнего обтекания непроницаемых и проницаемых поверхностей сжимаемыми газовыми потоками. На ее основе проведены многочисленные исследования самых различных турбулентных течений, которые показали хорошую точность и универсальность модели и позволили обнаружить в исследованных течениях ряд интересных особенностей и эффектов [1]. Данная методика была использована в расчетах перехода к развитой турбулентности в ПС и каналах при большом уровне внешних возмущений. Правильно описан характер перехода и с приемлемой для практики точностью получена количественная информация как по интегральным, так и по локальным характеристикам перехода.

Для течений в ПС с разнообразными граничными условиями получены экспериментально подтвержденные результаты расчетов в широком диапазоне параметров, вплоть до экстремальных (отсос и ускорение потока — до ламинаризации, торможение — до отрыва ПС), что имеет важное значение для практики и свидетельствует об универсальности используемой модели турбулентности.

Для течений в каналах переменного сечения использована описанная в работе [1] методика расчета, дополненная блоком определения продольного градиента давления [6].

1.1. Основные уравнения

Для решения задач использовались уравнения неразрывности, движения и энергии, описывающие существенно дозвуковое (число Маха M << 1) течение в плоском ($\alpha = 0$) или круглом ($\alpha = 1$) симметричном канале:

$$\frac{\partial}{\partial x}(\rho u) + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left(r^\alpha \rho v\right) = 0,$$

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial r} = -\frac{dp}{dx} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left[r^\alpha\left(\eta\frac{\partial u}{\partial r} + \rho\tau\right)\right], \qquad (1)$$

$$c_p\left(\rho u\frac{\partial T}{\partial x} + \rho v\frac{\partial T}{\partial r}\right) = u\frac{dp}{dx} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left[r^\alpha\left(\lambda\frac{\partial T}{\partial r} + \rho q_T\right)\right].$$

Здесь x — осевая координата, отсчитываемая от входа в канал; r — радиальная координата, отсчитываемая от геометрической оси канала; u и v — компоненты скорости вдоль осей x и r соответственно; p — давление; T — температура; $\rho\tau = -\rho\langle u'v'\rangle$ — турбулентное трение; $\rho q_T = -\rho c_p\langle vT'\rangle$ — турбулентный поток теплоты; ρ — плотность, η — динамическая вязкость; c_p — изобарная теплоемкость; λ — теплопроводность.

1.2. Модель турбулентности

Для вычисления значений τ и q_T в формуле (1) используем трехпараметрическую модель турбулентности [2], обобщенную на течение с теплообменом [7], в которой уравнения переноса записываются для энергии турбулентности $E = 0,5\sum_{i=1}^{2}\langle u_i'^2\rangle$, напряжения сдвига $\tau = -\langle u'v'\rangle$ и предложенного А. Н. Колмогоровым параметра $\omega = E/L^2$ (L — поперечный интеграль-

ный масштаб турбулентности), а также уравнение переноса для величины $q_T = -c_p \langle v'T' \rangle$ [8]:

$$\rho u \frac{\partial E}{\partial x} + \rho v \frac{\partial E}{\partial r} =$$
$$= -(c\rho\sqrt{E}L + c_1\eta)\frac{E}{L^2} + \rho\tau\frac{\partial u}{\partial r} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left(r^\alpha D_E \frac{\partial E}{\partial r}\right),$$
$$\rho u \frac{\partial \tau}{\partial x} + \rho v \frac{\partial \tau}{\partial r} =$$
$$= -(3c\rho\sqrt{E}L + 9c_1\eta)\frac{\tau}{L^2} + c_2\rho E\frac{\partial u}{\partial r} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left(r^\alpha D_\tau \frac{\partial \tau}{\partial r}\right) -$$
$$- \alpha D_\tau \frac{\tau}{r^2},$$
$$\rho u \frac{\partial \omega}{\partial x} + \rho v \frac{\partial \omega}{\partial r} = -(2c\rho\sqrt{E}L + 1,4c_1\eta f_\omega)\frac{\omega}{L^2} +$$
$$+\left[\frac{\tau}{E} - 2c_3\,\mathrm{sign}\left(\frac{\partial u}{\partial r}\right)\right]\rho\omega\frac{\partial u}{\partial r} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left(r^\alpha D_\omega \frac{\partial \omega}{\partial r}\right), \quad (2)$$
$$\rho u \frac{\partial q_T}{\partial x} + \rho v \frac{\partial q_T}{\partial r} = -[3c\rho\sqrt{E}L + 9c_1\eta f(\mathrm{Pr})]\frac{q_T}{L^2} +$$
$$+ c_4 c_p \rho E\frac{\partial T}{\partial r} + \frac{1}{r^\alpha}\frac{\partial}{\partial r}\left(r^\alpha D_q \frac{\partial q_T}{\partial y}\right) - \alpha D_q \frac{q_T}{r^2},$$
$$D_\varphi = a_\varphi\sqrt{E}L + \alpha_\varphi\eta\ (\varphi = E,\ \tau,\ \omega,\ q_T),\quad L = \sqrt{E/\omega},$$
$$f_\omega = 1 - \frac{1}{2c_1}\left(\frac{L}{E}\frac{\partial E}{\partial r}\right)^2,\quad f(\mathrm{Pr}) = \frac{1+c_5}{2}\frac{\sqrt{\mathrm{Pr}} + 1/\sqrt{\mathrm{Pr}}}{1 + c_5\sqrt{\mathrm{Pr}}}.$$

Значения констант [2, 7, 8]: $c = 0{,}3$; $c_1 = 5\pi/4$; $c_2 = 0{,}2$; $c_3 = 0{,}04$; $c_4 = 0{,}235$; $c_5 = 0{,}25$; $a_E = a_\omega = 0{,}06$; $a_\tau = a_q = 3a_E = 0{,}18$; $\alpha_E = \alpha_\tau = 1$; $\alpha_\omega = 1{,}4$; $\alpha_{q_T} = f(\mathrm{Pr})$. Pr — молекулярное число Прандтля.

Теплофизические свойства среды $(\rho, \eta, \lambda, c_p)$ в зависимости от температуры и давления задавались в табличном виде.

Граничные условия на стенках, на оси симметрии и на входе в круглый (плоский) канал будут формулироваться для каждой из рассмотренных ниже задач.

Система уравнений (1), (2) с соответствующими граничными условиями решалась численно методом прогонки с итерациями. Расчеты проводились на неравномерной сетке. Шаг по координате y определялся соотношениями

$$y = \delta\left[1 - \mathrm{cth}\,Q\,\mathrm{th}\left(Q(1-\varsigma)\right)\right],\quad \Delta y = \frac{\delta Q\,\mathrm{cth}\,Q\Delta\varsigma}{\mathrm{ch}^2\,Q(1-\varsigma)}$$
$$\left(0 \leqslant \varsigma \leqslant 1,\ \Delta\varsigma = \mathrm{const}\right).$$

Здесь δ — толщина ПС, определяемая с помощью соответствующего текущей задаче граничного условия при $y = \delta$ из условия гладкого сопряжения решений. Выбирая значение Q, можно добиться того, чтобы шаг Δy вблизи стенки был малым. Шаг по x был достаточно мал в сечениях, близких к входному сечению, и увеличивался по мере продвижения вниз по потоку.

2. Результаты численного моделирования

Ниже представлены результаты исследований смешанной конфекции в вертикальных трубах в условиях устойчивой и неустойчивой стратификации, а также течений в каналах с проницаемыми стенками при наличии вдува или отсоса на стенке.

2.1. Смешанная конвекция в вертикальных трубах

Взаимодействие поля силы тяжести и среды с переменной плотностью приводит к появлению сил плавучести (сил Архимеда), которые являются причиной возникновения движения в случае свободной конвекции и оказывают воздействие на вынужденное движение среды при смешанной конвекции. Если движение является турбулентным и пульсации плотности среды скоррелированы с пульсациями скорости, то сила тяжести может оказать влияние непосредственно и на характеристики турбулентности.

Исследованию влияния сил плавучести на среднее и пульсационное движение сред в природных явлениях и в технике посвящено достаточно много работ. Так, обзор результатов прикладных исследований содержится в работе [12]. Отметим, что число экспериментальных работ невелико. Это объясняется трудоемкостью организации и проведения эксперимента, связанной с необходимостью иметь рабочие участки большой длины и диаметра для обеспечения развитого течения, достижения достаточно больших чисел Грасгофа и возможности проводить измерения локальных характеристик потока.

Ниже представлены результаты численного исследования случая смешанной конвекции при течении в круглых вертикальных трубах, когда направление вектора силы тяжести совпадает с осью трубы. В такой постановке выполнена методическая часть работы, базирующаяся на трехпараметрической модели турбулентности [2, 7] с уравнением переноса для поперечного потока теплоты [8], которые дополнены термогравитационными членами.

2.1.1. Смешанная конвекция в условиях устойчивой стратификации

Для тестирования методики расчета привлечен экспериментальный материал [10—14] по устойчивому распределению плотности, имеющему место при восходящем течении в вертикальных обогреваемых трубах. Такой выбор экспериментов обусловлен тем, что этот случай исследован более детально, нежели случай неустойчивого распределения плотности, соответствующего нисходящему течению. В экспериментах [10—14] в качестве теплоносителя использовался воздух. Переменность плотности обеспечивалась нагреванием среды через стенку трубы.

Наиболее информативной представляется работа [13], в которой систематизированы результаты по исследованию влияния сил плавучести на интегральные и локальные характеристики течения в вертикальных обогреваемых трубах, полученные авторами ранее, а также содержатся новые результаты.

Проведенный анализ экспериментальных работ [10—14] ввиду различия полученных в них результатов не дает оснований ограничиться выбором одной из них для сравнения с расчетами. Поэтому сравнение проводилось с различными экспериментальными данными, а также с привлечением опытных данных, полученных при слабом влиянии сил плавучести.

Из расчетных моделей, базирующихся на уравнениях переноса для характеристик турбулентности, отметим работы [15, 16], в которых использовалась k-ε-модель турбулентности и восьмипараметрическая модель. Представляется, что двух параметров, используемых в работе [15], для рассматриваемого класса задач, по-видимому мало, поскольку при этом непосредственно не учитывается влияние термогравитации на перенос импульса и теплоты, а восемь параметров в работе [16] — много, так как экспериментальной информации, необходимой для апробирования модели и задания граничных условий в расчете, недостаточно.

Использование четырех параметров, уравнения переноса для которых приведены в работах [2, 7, 8], представляется оптимальным для описания основных особенностей рассматриваемой задачи, результаты решения которой приведены в работе [17].

Численное исследование смешанной конвекции в условиях неустойчивой стратификации, которая имеет место при опускном течении в обогреваемой трубе, приведено в работе [18]. Этот случай в экспериментальном плане (см., например, обзоры [9, 19]) исследован в значительно меньшей степени, нежели случай устойчивой стратификации. Так, данные измерений характеристик турбулентности отсутствуют. Имеются в основном результаты измерений коэффициента теплоотдачи. Расчетных работ, посвященных рассматриваемому случаю, в которых используются уравнения переноса для характеристик турбулентности, также немного [9, 19].

При решении задач смешанной конвекции в вертикальных трубах уравнение движения системы (1) должно быть дополнено членом, учитывающим влияние силы тяжести вида $\rho k_x g$, где g — модуль вектора ускорения силы тяжести; k_x — компонента единичного вектора ускорения силы тяжести в направлении оси x ($k_x = 1$, если направления движения и ускорения силы тяжести совпадают — опускное течение; $k_x = -1$, если они противоположны — подъемное течение). При движении среды с нагревом в поле силы тяжести уравнения модели турбулентности (2) также должны быть дополнены термогравитационными членами:

- уравнение для энергии турбулентности E — членом $+k_x g\langle\rho' u'\rangle$;
- уравнение для напряжения сдвига $\tau = -\langle u'v'\rangle$ — членом $-k_x g\langle\rho' v'\rangle$;
- уравнение для параметра $\omega = E/L^2$ — членом $+k_x g\langle\rho' u'\rangle/L^2$.

В предположении, что среда подчиняется уравнению состояния идеального газа с $\rho \sim 1/T$, корреляции пульсаций плотности и скорости могут быть аппроксимированы следующими соотношениями [9, 20]:

$$\langle\rho' u'\rangle = -\beta\rho\langle u'T'\rangle = -\frac{\beta\rho q_x}{c_p},$$

$$\langle\rho' v'\rangle = -\beta\rho\langle v'T'\rangle = -\frac{\beta\rho q_T}{c_p},$$

$$q_x = c_p\langle u'T'\rangle.$$

Здесь $\beta = 1/T$ — коэффициент объемного расширения газа; ρq_x — продольный турбулентный тепловой поток.

Для параметра q_x может быть записано уравнение переноса, аналогичное уравнению переноса для поперечного турбулентного потока теплоты q_T (2), и получена связь между продольным и поперечным потоками теплоты вида [17]

$$q_x = -\frac{q_T}{c_4}\left(2\frac{\tau}{E} + \frac{k_x g\beta}{c}\frac{q_T}{c_p}\frac{L}{E^{3/2}}\right).$$

При решении задачи о смешанной конвекции для замыкания системы уравнений (1)—(2) использовались следующие граничные условия на стенке, на оси симметрии и на входе в обогреваемый участок трубы диаметром d:

$$\begin{gathered}
u = 0,\ v = 0,\ T = \vartheta_0 \text{ или } -\left(\lambda\frac{\partial T}{\partial r}\right)_w = Q_0,\\
E = \frac{\partial E}{\partial r} = \tau = q_T = 0\ (r = d/2),\ \frac{\partial u}{\partial r} = v = 0,\\
\frac{\partial T}{\partial r} = 0,\ \frac{\partial E}{\partial r} = \frac{\partial\omega}{\partial r} = \tau = q_T = 0\ (r = 0),\\
p = p_1,\ u = u(r),\ T = T_0,\ E = E(r),\\
\omega = \omega(r),\quad \tau = \tau(r),\ q_T = 0\ (x = 0).
\end{gathered} \tag{3}$$

Здесь ϑ_0 и Q_0 — заданные температура или тепловой поток на стенке трубы.

Таким образом, система уравнений (1)—(2) с граничными условиями (3) позволяет решить задачу и найти распределения как средних, так и турбулентных характеристик течения и теплообмена.

Представленные в работе [17] результаты численного исследования для числа Нуссельта Nu, полей скорости и температуры, профилей турбулентного трения $\langle u'v'\rangle_+ = \langle u'v'\rangle/u_*$ и поперечного потока теплоты $\langle v'T'\rangle_+ = \langle v'T'\rangle/u_*T_*$ (рис. 1) сравниваются с экспериментальными данными. Показано, что зависимость числа Нуссельта Nu от x имеет минимум, который с ростом числа Грасгофа смещается к началу участка обогрева. Характер изменения профилей скорости и характеристик турбулентности свидетельствует о ламинаризации течения в области минимума зависимости Nu от x, после прохождения которого течение вновь турбулизуется. Результаты расчета в качественном отношении согласуются с имеющимися экспериментальными данными. Что касается количественного соответствия, то оно имеет место не для всех экспериментов, результаты которых также различаются между собой.

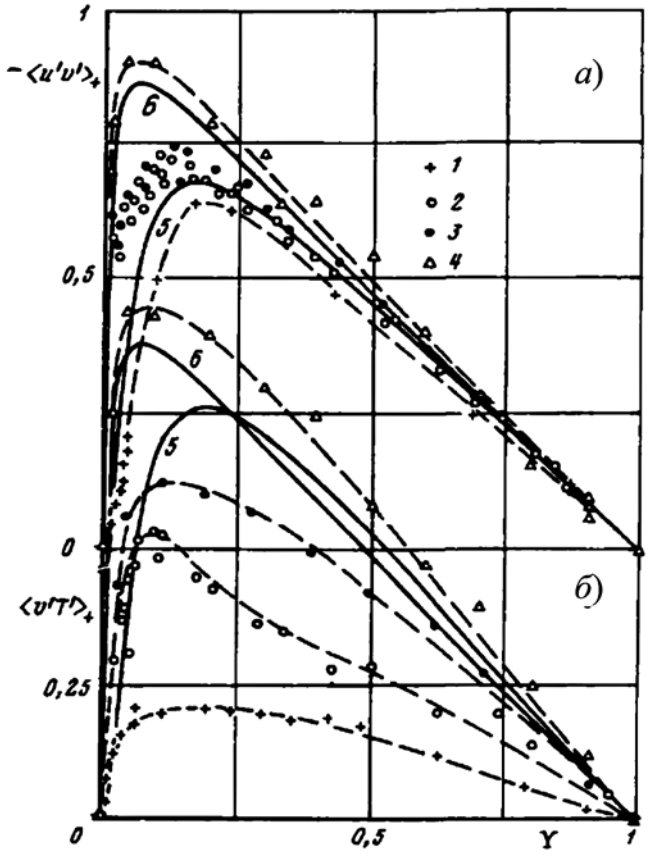

Рис. 1. Профили $\langle u'v'\rangle_+$ (*а*) и $\langle v'T'\rangle_+$ (*б*) по радиусу трубы $Y = y/r_0$: линии и точки — экспериментальные данные [20] для числа Рейнольдса Re = (5, 10—20, 40)·10³ (линии и точки *1—3* соответственно) при x/d = 53и данные [21] для Re = 32·10³ (линии и точки *4*) при x/d = 35; линии — результаты расчета для Re = 5·10³ (линия *5*) и Re = 40·10³ (линия *6*) на длине x/d = 50

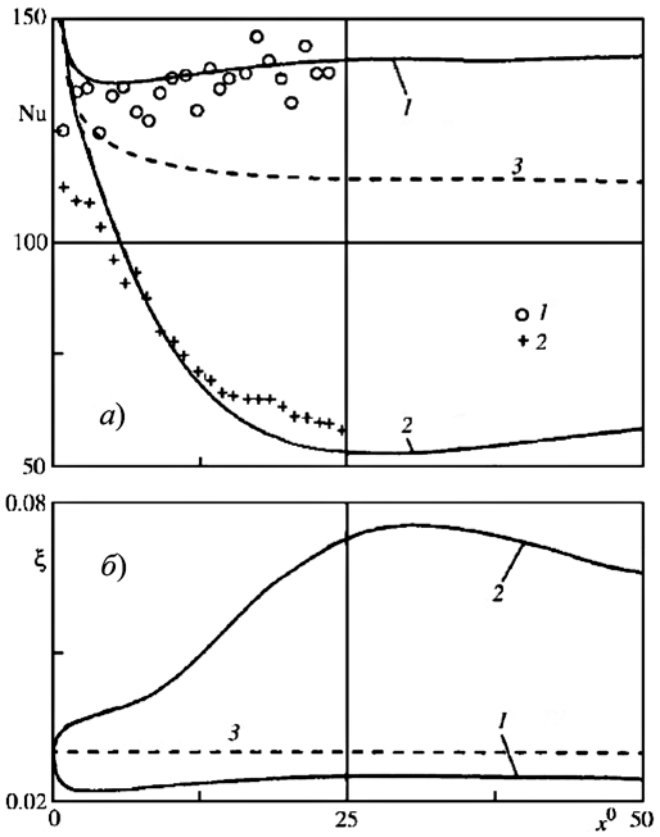

Рис. 2. Изменение числа Нуссельта Nu (*а*) и коэффициента сопротивления (*б*) по длине обогреваемого участка трубы для опускного (линии и точки *1*) и подъемного (линии и точки *2*) течения воды при значениях чисел Рейнольдса Re_0 = = 12·10³, Прандтля Pr_0 = 7, Грасгофа Gr_0 = 3,5·10⁹ и Архимеда Ar_0 = 9,8·10⁹, а также при отсутствии влияния сил плавучести (линия *3*); линии — расчет, точки — эксперимент [22]

2.1.2. Смешанная конвекция в условиях неустойчивой стратификации

В работе [18] проведено численное исследование смешанной конвекции в условиях неустойчивой стратификации, реализуемой при опускном течении в обогреваемой трубе в поле силы тяжести. Результаты расчета согласуются как с имеющимися опытными данными по локальной теплоотдаче, так и с обобщенными критериальными зависимостями. В расчете получены локальные характеристики течения и теплообмена, свидетельствующие о турбулизации потока, которые существенно отличаются от аналогичных характеристик для случая устойчивой стратификации, реализуемого при подъемном течении.

На рис. 2 представлено изменение по длине обогреваемого участка трубы $x^0 = x/d$ числа Нуссельта Nu (*а*) и коэффициента сопротивления ξ (*б*) для одного из режимов [22] с сильной стратификацией. Для сравнения на рис. 2, наряду с результатами расчета и экспериментальными данными для опускного течения (линии и точки *1*), представлены также расчетные экспериментальные данные при тех же значениях параметров Re_0, Pr_0, Gr_0, Ar_0 для подъемного течения (линии и точки *2*) и расчет (линии *3*) для случая отсутствия влияния сил плавучести (Gr → 0).

Как видно на рис. 2, влияние сил плавучести в случае опускного течения (неустойчивая стратификация) меньше, нежели в случае подъемного (устойчивая стратификация) по сравнению со случаем отсутствия сил плавучести. Это относится как к теплообмену, так и к сопротивлению при течении в трубе, для которого экспериментальные данные отсутствуют.

Проведенное численное исследование смешанной конвекции в условиях устойчивой и неустойчивой стратификации, реализуемой при подъемном и опускном течении в обогреваемой трубе в поле силы тяжести, позволило установить следующее.

Результаты расчета согласуются как с имеющимися опытными данными по локальным характеристикам течения и теплообмена, так и с обобщенными критериальными зависимостями. В случае неустойчивой стратификации, реализуемой при опускном течении, характеристики течения и теплообмена свидетельствуют о турбулизации потока и существенно отличаются от аналогичных характеристик для случая устойчивой стратификации, реализуемого при подъемном течении.

2.2. Влияние отсоса и вдува на течение в трубе с проницаемыми стенками

Известно [23], что существенное влияние на характеристики течения оказывают граничные условия на поверхности обтекаемых тел, например, вдув или отсос газа через проницаемую стенку. При этом результаты для внутреннего и внешнего обтеканий будут заметно отличаться. Так, при обтекании пластины безградиентным потоком с ростом интенсивности отсоса имеет место ламинаризация ПС [24]. При течении в трубе с отсосом газа, как следует из обзора экспериментальных работ [25], результаты зависят не только от интенсивности отсоса, но и от длины участка отсоса, причем эта зависимость неоднозначная. Это, по-видимому, связано с тем обстоятельством, что при отсосе газа скорость потока (и число Рейнольдса) по длине трубы падает, что приводит к возникновению положительного градиента давления. Это существенно влияет не только на интегральные, но и на локальные характеристики течения, в том числе на характеристики турбулентности потока.

Из упомянутых в обзоре [25] экспериментальных работ наиболее информативными представляются работы [26, 27]. В работе [26] экспериментальное исследование проведено на коротких участках (не более 10 калибров трубы) с интенсивным отсосом. В работе [27] измерения выполнены в широком диапазоне длин участков отсоса (до 50 калибров трубы), но при слабом отсосе. Представленные в работах [26, 27] результаты свидетельствуют о том, что в зависимости от интенсивности и длины участка отсоса наблюдается как уменьшение, так и увеличение уровня турбулентности в приосевой области. При сильном отсосе профиль скорости вытягивается у оси, а степень турбулентности существенно возрастает.

Из расчетных работ, рассмотренных в обзоре [25], заслуживает внимания работа [28], где на основе дифференциальной модели турбулентности проведены расчеты и предпринята попытка объяснить имеющиеся экспериментальные данные.

Использование проницаемых поверхностей в различных устройствах обусловлено необходимостью теплозащиты стенок с помощью вдува. Об актуальности проблемы теплозащиты стенок можно судить по большому числу публикаций по исследованию ПС с вдувом, достаточно полная библиография которых содержится в [23]. Экспериментальных работ по исследованию вдува при течении в трубах значительно меньше [25]. Как наиболее представительные следует отметить экспериментальные работы [29, 30], в которых получены результаты исследования турбулентного течения и теплообмена в пористой круглой трубе с равномерным вдувом газа через стенку. Из расчетных работ, посвященных исследованию турбулентного течения в круглой трубе с равномерным вдувом через пористые стенки, обзор которых представлен в работе [25], следует отметить работу [31], в которой использована трехпараметрическая модель турбулентности, представляющая собой один из вариантов модели [2].

При решении задач о течении в трубах с проницаемыми стенками для замыкания системы уравнений (1)—(2) использовались следующие граничные условия.

На стенке ($r = r_0$), на оси ($r = 0$) трубы и на входе ($x = 0$) в участок отсоса или вдува:

$$u = 0, \quad \rho v = (\rho v)_w, \; -\left(\lambda \frac{\partial T}{\partial r}\right)_w = q_w \;\text{ для отсоса},$$

$$\rho v = -(\rho v)_w, \; T = T_w \;\text{ для вдува},$$

$$E = \frac{\partial E}{\partial r} = \tau = 0 \;\; (r = r_0), \; \frac{\partial u}{\partial r} = v = 0,$$
$$\frac{\partial E}{\partial r} = \frac{\partial \omega}{\partial r} = \tau = 0 \;\; (r = 0), \qquad (4)$$
$$p = p_0, \; u = u(r), \; T = T_0, \; E = E(r),$$
$$\tau = \tau(r), \; \omega = \omega(r) \;\; (x = 0).$$

В качестве граничных условий на входе в участок отсоса (вдува) используются профили скорости и характеристик турбулентности для развитого течения в трубе при соответствующем числе Рейнольдса.

Параметрами задачи являются:

— число Рейнольдса на входе в участок отсоса (вдува) $\mathrm{Re}_0 = \rho d u_0/\eta$, где $d = 2r_0$, u_0 — средняя (по сечению) скорость на входе в участок отсоса (вдува);

— интенсивность отсоса (вдува) $j_{w0} = \pm(\rho v)_w/(\rho u)_0$, где $\pm(\rho v)_w(x)$ — заданная по длине массовая скорость отсоса (вдува).

Таким образом, система уравнений (1)—(2) с граничными условиями (4) при заданных параметрах Re_0 и j_{w0} позволяют решить задачу и найти распределения как средних, так и турбулентных характеристик течения в трубе с отсосом (вдувом).

2.2.1. Влияние отсоса на течение в трубе с проницаемыми стенками

Результаты [32—34] численного исследования течения и характеристик турбулентности в трубе с отсосом в широком диапазоне интенсивности отсоса и протяженности участка отсоса позволили установить следующее.

Для ряда значений интенсивности отсоса j_{w0} при $\mathrm{Re}_0 = 21\,710$ получено (рис. 3) изменение давления $\Delta P = (p - p_0)/\rho u_0^2$ по длине участка отсоса. Как видно, результаты расчета (линии) вполне удовлетворительно

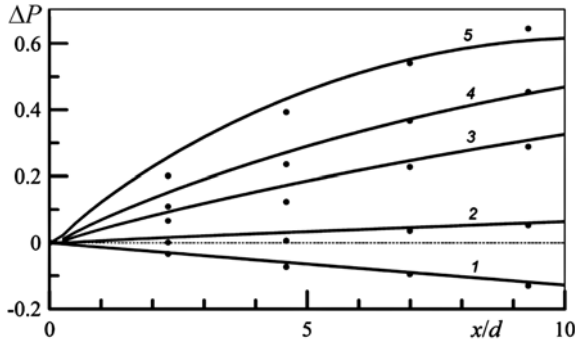

Рис. 3. Изменение давления ΔP по длине участка отсоса x/d для ряда значений величины интенсивности отсоса j_{w0}:
1—5 при j_{w0} = 0, 0,0027, 0,008, 0,0135, 0,024 соответственно; линии — расчет, точки — эксперимент [26]

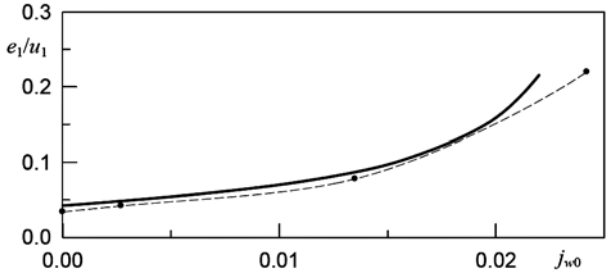

Рис. 4. Зависимости величины e_1/u_1 от интенсивности отсоса j_{w0} ($\mathrm{Re}_0 = 10^5$, $x/d = 9{,}5$):
линии — расчет, точки — эксперимент [26]

согласуются с экспериментальными данными [26] (точки) в широком диапазоне значений j_{w0} и свидетельствуют о существенном росте давления при интенсивном отсосе.

Показано, что при слабом отсосе, как при постоянной, так и при убывающей (пропорционально числу Рейнольдса) по длине интенсивности отсоса, профили скорости уплощаются, интенсивность турбулентности в большей части сечения трубы уменьшается, а на оси трубы она сначала возрастает, а затем убывает по длине.

Изменение интенсивности турбулентности на оси трубы e_1/u_1 $\left(e = \sqrt{E}\right)$ по длине участка отсоса (рис. 4) свидетельствует о том, что при слабом ($j_{w0} < 0{,}005$) отсосе значение e_1 на расчетной длине практически не возрастает и только при сильном ($j_{w0} > 0{,}01$) отсосе, как и в эксперименте [26], наблюдается рост интенсивности турбулентности на оси трубы.

При сильном постоянном по длине отсосе с ростом интенсивности отсоса профили скорости вытягиваются, интенсивность турбулентности возрастает, т. е. имеет место турбулизация потока. При этом средняя по сечению скорость потока (число Рейнольдса) существенно падает, а длина участка отсоса сокращается.

Полученные результаты численного исследования качественно соответствуют экспериментальным данным [26, 27], для достижения количественного согласования с экспериментом необходимо учесть ряд особенностей экспериментов [26, 27], в частности, значительную шероховатость стенок трубы, условия на входе в участок отсоса и др. Поэтому, ввиду значительного числа параметров задачи, представление результатов в виде обобщенных критериальных зависимостей затруднительно и до настоящего времени отсутствует. При необходимости для конкретных условий задачи результаты могут быть получены с использованием методики расчета, изложенной в настоящей работе.

2.2.2. Влияние вдува на течение в трубе с проницаемыми стенками

Представленные в работе [33] результаты численного исследования течения и теплообмена в трубе с проницаемыми стенками в зависимости от интенсивности вдува охладителя через стенки трубы позволило установить следующее.

Профили скорости в ряде сечений по длине (x/d, где x отсчитывается от входа в пористую трубу) при $\mathrm{Re} = 80\,000$, $j_{w0} = 0{,}012$ (рис. 5) за счет вдува становятся более вытянутыми по длине трубы, что согласуется с экспериментом [29]. Расчетные зависимости относительного коэффициента трения C_f/C_{f0} (C_{f0} — значение C_f при $j_{w0} = 0$) от интенсивности вдува j_{w0} для $\mathrm{Re}_0 = 28\,000$ и $80\,000$ (рис. 6) хорошо согласуются с экспериментальными данными [29] и свидетельствуют о существенном снижении коэффициента трения при вдуве.

Численное исследование теплообмена в пористой трубе со вдувом проведено для условий, соответствующих экспериментальным [30]. В экспериментах [30]

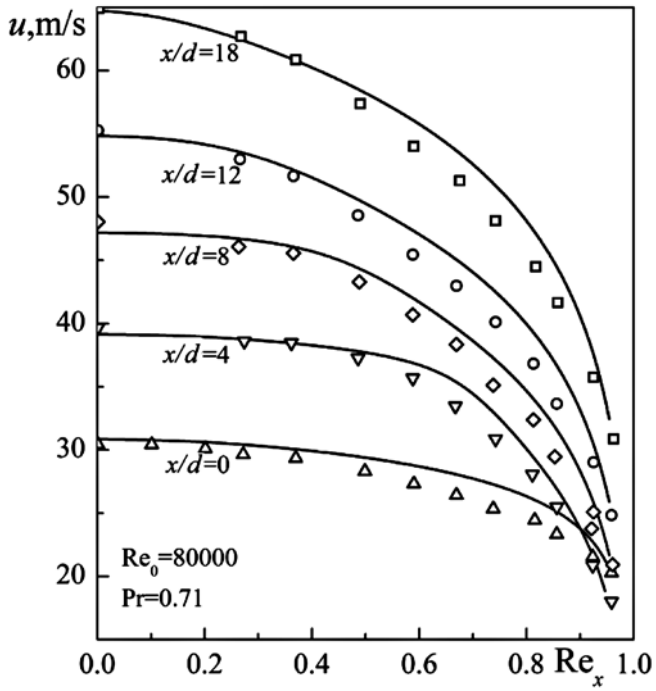

Рис. 5. Профили скорости для ряда сечений по длине трубы x/d для $j_{w0} = 0{,}012$:
линии — расчет, точки — эксперимент [29]

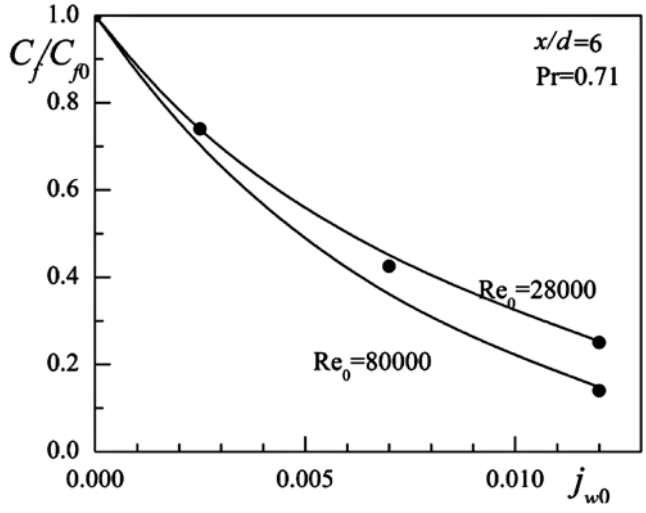

Рис. 6. Зависимость относительной величины коэффициента трения от интенсивности вдува в сечении по длине трубы $x/d = 6$ для двух чисел Рейнольдса:
линии — расчет, точки — эксперимент [29]

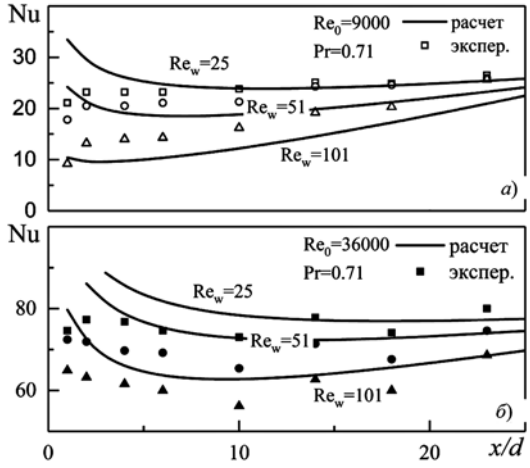

Рис. 7. Изменение числа Нуссельта Nu по длине участка вдува трубы.

Точки — эксперимент [30], линии — расчет: *а* — $\mathrm{Re}_0 = 9000$; *б* — $\mathrm{Re}_0 = 36\,000$; *1* — $\mathrm{Re}_w = 25$; *2* — $\mathrm{Re}_w = 51$; *3* — $\mathrm{Re}_w = 101$

гидродинамически развитый поток воздуха при температуре $T_0 = 300$ K поступал во входной участок трубы, а нагретый до $T = 330$ K воздух вдувался через стенку трубы на участке относительной длины $L/d \cong 24$. Эксперименты проводили для значений чисел Рейнольдса основного потока на входе $\mathrm{Re}_0 = (4...67)\cdot 10^3$ и для чисел Рейнольдса вдува $\mathrm{Re}_w = j_{w0}\,\mathrm{Re}_0 = 25,\ 51,\ 101,\ 197$.

Результаты расчетов безразмерного коэффициента теплоотдачи (числа Нуссельта) Nu и сравнение с экспериментальными данными [30] представлено на рис. 7. Как видно на рис. 7, согласование результатов расчета с экспериментом [30] зависимости $\mathrm{Nu}(x/d)$ при $x/d > 10$ можно считать удовлетворительным, что подтверждено также обработкой расчетных и экспериментальных данных зависимостью числа Нуссельта Nu от локального числа Рейнольдса Re_x.

Проведенное сравнение результатов расчета, отражающих основные особенности достаточно сложных процессов перестройки течения в трубе при вдуве газа, с известными экспериментальными данными в широком диапазоне чисел Рейнольдса и интенсивности вдува, показало их удовлетворительное согласование. Этот факт дает основание для использования разработанной методики расчета при проведении численного исследования течения и теплообмена в целях определения основных параметров, характеризующих эффективность теплообменных аппаратов с пористым охлаждением.

Заключение

С использованием трехпараметрической дифференциальной модели турбулентности разработана методика численного исследования в приближении УК задач течения и теплообмена в плоских и круглых каналах переменного сечения. Представлено описание вычислительной методики, приведены основные уравнения, характеризующие течение в каналах, а также уравнения и константы используемой авторами модели турбулентности. В части 1 обзорного исследования описаны основные результаты численного моделирования с использованием предложенной методики для решения задач о смешанной конвекции в вертикальных трубах, а также влиянии вдува и отсоса на течение в трубе.

Проведенное численное исследование смешанной конвекции в условиях устойчивой и неустойчивой стратификации, реализуемой при подъемном и опускном течении в обогреваемой трубе в поле силы тяжести, позволило установить следующее. Результаты расчета согласуются с имеющимися опытными данными как по локальным характеристикам течения и теплообмена, так и с обобщенными критериальными зависимостями. В случае неустойчивой стратификации, реализуемой при опускном течении, характеристики течения и теплообмена свидетельствуют о турбулизации потока и существенно отличаются от аналогичных характеристик для случая устойчивой стратификации, реализуемого при подъемном течении.

Путем численного моделирования турбулентного течения в трубе получены зависимости характеристик течения и турбулентности от интенсивности отсоса. Проведенное сравнение результатов расчета с известными экспериментальными данными показало их качественное соответствие, отражающее основные особенности достаточно сложных процессов перестройки течения при отсосе в трубе. При вдуве получены зависимости характеристик течения и теплообмена от интенсивности вдува охладителя через пористые стенки трубы. Показано, что коэффициенты трения и теплоотдачи уменьшаются с ростом интенсивности вдува. Проведенное сравнение результатов расчета с известными экспериментальными данными в широком диапазоне чисел Рейнольдса и интенсивности вдува показало их удовлетворительное согласование. Это дает основание для использования разработанной методики расчета при проведении численного исследования течения и теплообмена в целях определения основных параметров, характеризующих эффективность теплообменных аппаратов с пористым охлаждением.

Работа выполнена в рамках исследований, частично финансируемых гос. бюджетом (АААА-А16-116021110203-6, АААА-А19-119012990110-0; АААА-А16-116021110200-5; АААА-А19-119012990115-5) и Советом по грантам Президента РФ (№ СП-3993.2018.1).

Список литературы

1. **Лущик В. Г., Макарова М. С., Якубенко А. Е.** Применение трехпараметрической модели сдвиговой турбулентности для решения задач внешнего обтекания проницаемых поверхностей потоком сжимаемого газа // Программная инженерия. — 2017. — Т. 8, № 12. — С. 563—574.

2. **Лущик В. Г., Павельев А. А., Якубенко А. Е.** Трехпараметрическая модель сдвиговой турбулентности // Изв. АН СССР. МЖГ. — 1978. — № 3. — С. 13—25.

3. **Лущик В. Г., Павельев А. А., Якубенко А. Е.** Уравнения переноса для характеристик турбулентности: модели и результаты расчетов // В сб.: Итоги науки и техники. Сер. Механика жидкости и газа. М.: ВИНИТИ. — 1988. — Т. 22. — С. 3—61.

4. **Лапин Ю. В., Стрелец М. Х.** Внутренние течения газовых смесей. — М.: Наука, 1989. — 368 с.

5. **Егоров Ю. Э., Стрелец М. Х., Шур М. Л.** Границы применимости параболических моделей для численного исследования течений в соплах Лаваля. I. Приближение узкого канала. Препринт № 4. — Л.: НПО ГИПХ, 1991.

6. **Лущик В. Г., Сизов В. И., Якубенко А. Е.** К использованию приближения узкого канала для расчета турбулентного течения в соплах жидкостных ракетных двигателей // ТВТ. — 1993. — Т. 31, № 5. — С. 752—758.

7. **Лущик В. Г., Павельев А. А., Якубенко А. Е.** Трехпараметрическая модель турбулентности: расчет теплообмена // Изв. АН СССР. МЖГ. — 1986. — № 2. — С. 40—52.

8. **Лущик В. Г., Павельев А. А., Якубенко А. Е.** Уравнение переноса для турбулентного потока тепла. Расчет теплообмена в трубе // Изв. АН СССР. МЖГ. — 1988. — № 6. — С. 42—50.
9. **Петухов Б. С., Поляков А. Ф.** Теплообмен при смешанной турбулентной конвекции. — М.: Наука, 1986. — 192 с.
10. **Steiner A.** On the reverse transition of a turbulent flow under the action of buoyancy forces // J. Fluid Mech. — 1971. — Vol. 47, Pt. 3. — P. 503—512.
11. **Carr A. D., Connor M. A., Buhr H. O.** Velocity, Temperature, and Turbulence Measurements in Air for Pipe Flow With Combined Free and Forced Convection // J. Heat Transfer. — 1973. — Vol. 95, No 4. — P. 445—452.
12. **Polyakov A. F., Shindin C. A.** Development of turbulent heat transfer over the length of vertical tubes in the present of mixed air convection // Int. J. Heat Mass Transfer. — 1988. — Vol. 31, No. 5. — P. 987—992.
13. **Поляков А. Ф., Шиндин С. А.** Турбулентный перенос импульса и тепла при восходящем течении воздуха в обогреваемых трубах // Турбулентный теплообмен при смешанной конвекции в вертикальных трубах. — М.: ИВТАН, 1989. — С. 49—94.
14. **Вилемас Ю. В., Пошкас П. С., Каупас В. Э.** Обобщение данных по локальному теплообмену в газоохлаждаемой вертикальной трубе при смешанной конвекции // Турбулентная конвекция. — М.: ИВТАН, 1990. — С. 49—68.
15. **Cotton M. A., Nott I. D.** Computation of developing turbulent mixed convection heat transfer to air in a vertical tube: Comparison of low-Reynolds-number k-$\acute{\varepsilon}$-turbulence model with recent experiments // Турбулентная конвекция.— М.: ИВТАН, 1990. — С. 69—94.
16. **Ушпурас Э. В., Пошкас П. С.** Численное исследование характеристик турбулентного переноса при смешанной конвекции в вертикальных трубах // Турбулентная конвекция. М.: ИВТАН, 1990. — С. 95—113.
17. **Лущик В. Г., Якубенко А. Е.** Дифференциальная модель турбулентности: численное исследование смешанной конвекции в вертикальных трубах // Изв. РАН. МЖГ. — 1996. — № 2. — С. 73—86.
18. **Лущик В. Г., Якубенко А. Е.** Численное исследование смешанной конвекции в вертикальных трубах в условиях неустойчивой стратификации // Изв. РАН. МЖГ. — 2002. — № 3. — С. 50—57.
19. **Jackson J. D., Cotton M. A., Axell B. P.** Studies of mixed convection in vertical tubes // Int. J. Heat and Fluid Flow. — 1989. — Vol. 10, No. 1. — P. 2—15.
20. **Поляков А. Ф., Шиндин С. А., Комаров П. Л.** Турбулентное течение воздуха в круглых трубах при малых числах Рейнольдса // Турбулентный теплообмен при смешанной конвекции в вертикальных трубах. — М.: ИВТАН, 1989. С. 25—48.
21. **Ибрагимов М. Х., Субботин В. И., Таранов Г. С.** Определение корреляционной связи между пульсациями скорости и температуры в турбулентном потоке воздуха в трубе // Докл. АН СССР. — 1968. — Т. 183, № 5. — С. 1032—1035.
22. **Jackson J. D., Cotton M. A., Yu L. S. L.** et al. Turbulent mixed convection heat transfer to water in a vertical tube. Comparison between experiment and theory // Турбулентная конвекция. — М.: ИВТАН, 1990. — С. 23—47.
23. **Кутателадзе С. С., Леонтьев А. И.** Тепломассообмен и трение в турбулентном пограничном слое. — М.: Энергоатомиздат, 1985. — 319 с.
24. **Леонтьев А. И., Лущик В. Г., Якубенко А. Е.** Пограничный слой на проницаемой пластине с отсосом газа // ТВТ. — 2010. — Т. 48, № 3. — С. 396—401.
25. **Ерошенко В. М., Зайчик Л. И.** Гидродинамика и тепломассообмен на проницаемых поверхностях. — М.: Наука, 1984. — 274 с.
26. **Aggarwal M. A., Hollingsworth M. A., Mayhew Y. R.** Experimental friction factor for turbulent flow with suction in a porous tube // Int. J. Heat Mass Transfer. 1972. — Vol. 15. — P. 1585—1602.
27. **Brosh A., Vinograd Y.** Experimental study of turbulent flow in a tube with wall suction // Trans. ASME J. Heat Transfer. Ser. C. — 1974. — Vol. 96. — P. 338—342.
28. **Ерошенко В. М., Ершов А. В., Зайчик Л. И.** Расчет турбулентного течения несжимаемой жидкости в круглой трубе с отсосом через пористые стенки // Изв. АН СССР. МЖГ. — 1982. — № 4. — С. 87—93.
29. **Олсон Т., Эккерт Д.** Экспериментальное исследование турбулентного течения в пористой круглой трубе с равномерным вдувом газа через стенку // Прикладная механика. — 1966. — Т. 88, № 1. — С. 7—20.
30. **Lombardi G., Sparrow E. M., Eckert E. R. G.** Experiments on heat transfer to transpired turbulent pipe flows // Int. J. Heat Mass Transfer. — 1974. — Vol. 17. — P. 429—437.
31. **Ерошенко В. М., Ершов А. В., Зайчик Л. И.** Турбулентное течение жидкости в круглой трубе с равномерным вдувом через пористые стенки // ИФЖ. — 1981. — Т. XLI, № 5. — С. 791—795.
32. **Леонтьев А. И., Лущик В. Г., Макарова М. С.** Численное исследование течения в трубе с отсосом газа через проницаемые стенки // Изв. РАН. МЖГ. — 2014. — № 3. — С. 74—81.
33. **Makarova M. S., Lushchik V. G.** Numerical simulation of turbulent flow and heat transfer in tube under injection of gas through permeable walls // Journal of Physics: Conference Series. — 2017. — Vol. 891, No. 1. — 012066.
34. **Leontev A. I., Lushchik V. G., Makarova M. S.** Heat and mass transfer in a tube with permeable walls: influence of suction and the Prandtl number // Proccedings of the Eight International Symposium on Turbulence, Heat and Mass Transfer. — 2015. — P. 145—147.

Application of the Three-Parameter Differential Model of Turbulence for Solving Problems of Flow and Heat Transfer in Channels of Variable Cross-Section. Part 1

V. G. Lushchik, vgl_41@mail.u, **M. S. Makarova**, mariia.makarova@gmail.com,
A. I. Reshmin, alexreshmin@rambler.ru, Lomonosov Moscow State University,
Institute of mechanics, Moscow, 119192, Russian Federation

Corresponding author:

Makarova Mariia S., Researcher, Lomonosov Moscow State University, Institute of mechanics, Moscow, 119192, Russian Federation
E-mail: mariia.makarova@gmail.com

Received on September 18, 2020
Accepted on October 13, 2020

A description of the method of numerical study in the approximation of a narrow channel of the problems of flow and heat transfer in flat and circular channels of variable cross-section using a differential three-parameter model of shear turbulence is presented. The main results of numerous studies using the proposed method are described, one of the goals of which was to substantiate the possibility of using the narrow channel approximation. This review study

is carried out in two parts. In the first part the results of studies of mixed convection in vertical pipes under conditions of stable and unstable stratification, as well as flows in channels with permeable walls in the presence of blowing or suction on the wall, are presented.

Keywords: *differential turbulence model, diffuser, confuser, pressure gradient, blowing, suction, laminarization of the flow, intensification of heat transfer*

Acknowledgements:

This work was partially supported by the state budget (AAAA-A16-116021110203-6, AAAA-A19-119012990110-0; AAAA-A16-116021110200-5; AAAA-A19-119012990115-5) and the Grant Council of the President of the Russian Federation (СП-3993.2018.1).

For citation:

Lushchik V. G., Makarova M. S., Reshmin A. I. Application of the Three-Parameter Differential Model of Turbulence for Solving Problems of Flow and Heat Transfer in Channels of Variable Cross-Section. Part 1, *Programmnaya Ingeneria,* 2020, vol. 12, no. 1, pp. 21—30

DOI: 10.17587/prin.12.21-30

References

1. **Lushchik V. G., Makarova M. S., Yakubenko A. E.** Application of the Three-Parameter Model of Shear Turbulence for Solving Problems of External Flowing on Permeable Surfaces by a Compressible Gas Flow, *Programmnaya Ingeneria,* 2017, vol. 8, no 12, pp. 563—574 (in Russian).

2. **Lushchik V. G., Pavel'ev A. A., Yakubenko A. E.** Three-parameter model of shear turbulence, *Fluid Dynamics,* 1978, vol. 13, no. 3, pp. 350—360.

3. **Lushchik V. G., Pavel'ev A. A., Yakubenko A. E.** The transport equations for turbulence characteristics: models and calculation results, *Itogi Nauki i Tekhniki. Ser. Mechanics of fluid and gas,* VINITI, Moscow, 1988, vol. 22, pp. 3—61 (in Russian).

4. **Lapin Yu. V., Strelets M. H.** *Internal flows of gas mixtures,* Moscow, Nauka, 1989, 368 p. (in Russian).

5. **Egorov Yu., Strelets M., Shur M.** et al. The limits of applicability of parabolic models for the numerical study of flows in Laval nozzles. I. Approximation of a narrow channel. Preprint no. 4. NPO GIPH. Leningrad, 1991 (in Russian).

6. **Lushchik V. G., Sizov V. I., Yakubenko A. E.** On using the narrow channel approximation for turbulent flow calculation in liquid propellant engine nozzles, *High Temperatur*e, 1993, vol. 31, no. 5, pp. 752—758.

7. **Lushchik V. G., Pavel'ev A. A., Yakubenko A. E.** Three-parameter model of turbulence: Heat transfer calculations, *Fluid Dynamics,* 1986, vol. 21, no. 2, pp. 200—211.

8. **Lushchik V. G., Pavel'ev A. A., Yakubenko A. E.** Transfer equation for turbulent heat flux. Calculation of heat transfer in a pipe, *Fluid Dynamics*, 1988, vol. 23, no. 6, pp. 835—842.

9. **Petukhov B. S., Polyakov A. F.** *Heat transfer under the mixed turbulent convection,* Moscow, Nauka, 1986, 192 p. (in Russian).

10. **Steiner A.** On the reverse transition of a turbulent flow under the action of buoyancy forces, *J. Fluid Mech.,* 1971, vol. 47, no. 3, pp. 503—512.

11. **Carr A. D., Connor M. A., Buhr H. O.** Velocity, Temperature, and Turbulence Measurements in Air for Pipe Flow With Combined Free and Forced Convection, *J. Heat Transfer*, 1973, vol. 95, no. 4, pp. 445—452.

12. **Polyakov A. F., Shindin C. A.** Development of turbulent heat transfer over the length of vertical tubes in the present of mixed air convection, *Int. J. Heat Mass Transfer,* 1988, vol. 31, no. 5, pp. 987—992.

13. **Polyakov A. F., Shindin S. A.** Turbulent transfer of momentum and heat during an ascending air flow in heated pipes, *Turbulentnyj teploobmen pri smeshannoj konvekcii v vertikal'nyh trubah*, JIHT RAS, Moscow, 1989, pp. 49—94 (in Russian).

14. **Vilemas Yu. V., Poshkas P. S., Kaupas V. E.** Generalization of data on local heat transfer in a gas-cooled vertical pipe with mixed convection, *Turbulentnaya konvekciya*, JIHT RAS, Moscow, 1990, pp. 49—68 (in Russian).

15. **Cotton M. A., Nott I. D.** Computation of developing turbulent mixed convection heat transfer to air in a vertical tube: Comparison of low-Reynolds-number *k*-ε-turbulence model with recent experiments, *Turbulentnaya konvekciya*, JIHT RAS, Moscow, 1990, pp. 69—94 (in Russian).

16. **Ushpuras E. V., Poshkas P. S.** Numerical study of the characteristics of turbulent transfer under mixed convection in vertical tubes, *Turbulentnaya konvekciya*, JIHT RAS, Moscow, 1990, pp. 95—113 (in Russian).

17. **Lushchik V. G., Yakubenko A. E.** Differential model of turbulence: A numerical study of mixed convection in vertical pipes, *Fluid Dynamics*, 1996, vol. 31, no 2, pp. 73—86.

18. **Lushchik V. G., Yakubenko A. E.** Numerical Study of Mixed Convection in Vertical Tubes under Conditions of Unstable Stratification, *Fluid Dynamics,* 2002, vol. 37, no. 3, pp. 50—57.

19. **Jackson J. D., Cotton M. A., Axell B. P.** Studies of mixed convection in vertical tubes, *Int. J. Heat and Fluid Flow,* 1989, vol. 10, no. 1, pp. 2—15.

20. **Polyakov A. F., Shindin S. A., Komarov P. L.** Turbulent air flow in round tubes at low Reynolds numbers, *Turbulentnyj teploobmen pri smeshannoj konvekcii v vertikal'nyh trubah*, Moscow, JIHT RAS, 1989, pp. 25—48 (in Russian).

21. **Ibragimov M. H., Subbotin V. I., Taranov G. S.** Determination of the correlation between velocity and temperature fluctuations in a turbulent air flow in a pipe, *Dokl. AN SSSR,* 1968, vol. 183. no. 5, pp. 1032—1035 (in Russian).

22. **Jackson J. D., Cotton M. A., Yu L. S. L.** et al. Turbulent mixed convection heat transfer to water in a vertical tube. Comparison between experiment and theory, *Turbulentnaya konvekciya*, Moscow, IJHT RAS, 1990, pp. 23—47.

23. **Kutateladze S. S., Leontiev A. I.** *Heat and Mass Transfer and Friction in a Turbulent Boundary Layer* (2nd edition), Moscow, Energoatomizdat, 1985, 319 p. (in Russian).

24. **Leontiev A. I., Lushchik V. G., Yakubenko A. E.** Boundary layer on a permeable wall with suction of gas, *High temperature*, 2010, vol. 48, no. 3, pp. 376—381.

25. **Eroshenko V. M., Zaichik L. I.** *Fluid Dynamics and Heat and Mass Transfer on Permeable Surfaces*, Moscow, Nauka, 1984, 275 p. (in Russian).

26. **Aggarwal M. A., Hollingsworth M. A., Mayhew Y. R.** Experimental friction factor for turbulent flow with suction in a porous tube, *Int. J. Heat Mass Transfer,* 1972, vol. 15, pp. 1585—1602.

27. **Brosh A., Vinograd Y.** Experimental study of turbulent flow in a tube with wall suction, *Trans. ASME J. Heat Transfer. Ser. C,* 1974, vol. 96, pp. 338—342.

28. **Eroshenko V. M., Ershov A. V., Zaichik L. I.** Calculation of turbulent flow of an incompressible fluid in a circular tube with suction through porous walls, *Fluid Dynamics,* 1982, vol. 17, no. 4, pp. 559—564.

29. **Olson R. M., Eckert E. R. G.** Experimental Studies of Turbulent Flow in a Porous Circular Tube With Uniform Fluid Injection Through the Tube Wall, *J. Applied Mechanics,* 1966, vol. 33, no. 1, pp. 7—17.

30. **Lombardi G., Sparrow E. M., Eckert E. R. G.** Experiments on heat transfer to transpired turbulent pipe flows, *Int. J. Heat Mass Transfer,* 1974, vol. 17, pp. 429—437.

31. **Eroshenko V. M., Ershov A. V., Zaichik L. I.** Turbulent fluid flow in a circular pipe with uniform blowing through porous walls, J. *Engineering Physics and Thermophysics,* 1981, vol. 41, no. 5, pp. 1175—1178.

32. **Leont'ev A. I., Lushchik V. G., Makarova M. S.** Numerical investigation of tube flow with suction through permeable walls, *Fluid Dynamics,* 2014, vol. 49, no. 3, pp. 362—368.

33. **Makarova M. S., Lushchik V. G.** Numerical simulation of turbulent flow and heat transfer in tube under injection of gas through permeable walls, *Journal of Physics: Conference Series*, 2017, vol. 891, no. 1, 012066.

34. **Leontiev A. I., Lushchik V. G., Makarova M. S.** Heat and mass transfer in a tube with permeable walls: influence of suction and the Prandtl number, *Proccedings of the Eight International Symposium on Turbulence, Heat and Mass Transfer,* 2015, pp. 145—147.

УДК 004.8, 004.93 DOI: 10.17587/prin.12.31-39

Д. Д. Рухович, инженер, d.rukhovich@samsung.com, Центр Искусственного Интеллекта Самсунг, Москва

Итеративный метод обнаружения объектов

Описан новый метод обнаружения объектов на цифровом изображении на основе глубокого обучения. Существующие нейросетевые модели обнаружения объектов показывают высокую точность в простых случаях, когда изображение содержит несколько крупных неперекрывающихся объектов. Однако если на изображении присутствует большое число перекрывающихся объектов, то подобные модели нередко пропускают часть объектов, так что итоговое число обнаруженных объектов оказывается значительно меньше истинного. В статье описан способ модификации нейросетевых моделей обнаружения объектов, позволяющий находить объекты на изображении за несколько итераций, при этом объекты, найденные на предыдущих итерациях, маскируются на текущем шаге. Предложенная модификация может быть применена к одно- и двухстадийным нейросетевым моделям обнаружения объектов. Экспериментальная проверка показала, что предложенная модификация позволяет повысить точность нейросетевых моделей обнаружения объектов для наборов данных, содержащих изображения скоплений объектов.

Ключевые слова: *глубокое обучение, обнаружение объектов, нейронные сети, метод подавления немаксимумов*

Введение

Обнаружение объектов — одна из основных задач компьютерного зрения. Она заключается в поиске, локализации и классификации объектов заданного типа на изображении. Методы обнаружения объектов принимают на вход изображение и возвращают неупорядоченный набор оценок местоположений объектов в виде прямоугольников, ограничивающих область на изображении, в которой находится объект. Далее эти прямоугольники будут называться рамками объекта. В последнее время активно развиваются методы обнаружения объектов, основанные на глубоком обучении [1—5].

Большинство современных нейросетевых моделей обнаружения объектов предсказывают несколько возможных рамок для каждого объекта. Затем все гипотезы о расположении объекта независимо оцениваются и из них выбирается единственная, таким образом, расположение каждого объекта определяется однозначно. Для этого применяется метод подавления немаксимумов (*non-maximum suppression*, NMS) или аналогичные ему "жадные" алгоритмы. Однако алгоритмы подобного рода работают эффективно только в тех случаях, когда объекты одного класса не перекрываются на изображении.

Обработка изображений с большим числом пересекающихся объектов одного класса (например, массовые скопления людей или колонии бактерий на снимках с микроскопа) представляет трудность для многих моделей обнаружения объектов. Наблюдаемый эффект обусловлен двумя причинами. Во-первых, при большой площади перекрытий рамок объектов сложно определить, сколько близко расположенных объектов находится внутри предсказанной алгоритмом локации. Во-вторых, при большом числе объектов на изображении каждый из объектов занимает небольшую площадь изображения. Иногда эта площадь настолько мала, что оказывается за пределами чувствительности метода. В последнее время был опубликован ряд работ, в которых исследуются модификации алгоритма подавления немаксимумов [6—11]. Несмотря на некоторое улучшение качественных характеристик относительного стандартного алгоритма подавления немаксимумов, все "жадные" алгоритмы имеют известные ограничения. При решении задачи отбора предсказаний необходимо удалять дубликаты, увеличивая точность, но сохранять предсказания, соответствующие перекрывающимся объектам, увеличивая полноту. Путем настройки гиперпараметров можно регулировать соотношение между точностью и полнотой метода, однако невозможно добиться улучшения сразу обоих показателей.

В статье описан способ модификации нейросетевых моделей для обнаружения объектов, позволяющий обнаруживать объекты не единовременно, а в несколько итераций, на каждой из которых определяется новое подмножество объектов. Все объекты, найденные на предыдущих итерациях, учитываются на последующих, таким образом удается избежать дублирования предсказаний. Предложенный способ модификации получил название итеративный детектор (*Iterative Detector* — IterDet). Этот способ может быть применен к любой нейросетевой модели обнаружения объектов. Это потребует незначительного изменения процедуры обучения и предсказания. Далее модели, модифицированные с помощью IterDet, будут указываться с суффиксом -IterDet.

На рис. 1 (см. третью сторону обложки) показаны результаты работы модели Faster RCNN-IterDet на изо-

бражении из тестовой части набора CrowdHuman [12]. На второй итерации найдены 9 объектов (показанные желтым цветом) из 137. Таким образом, по сравнению с Faster RCNN без модификации IterDet удалось дополнительно обнаружить пять объектов и повысить среднюю полноту предсказаний на 2,7 %.

В нескольких работах [13—16] были предложены альтернативные нейросетевые модели обнаружения объектов, предназначенные для изображений с большим числом пересекающихся объектов. Так, например, в работе [16] рассматривается сверточно-рекуррентная модель для итеративного обнаружения объектов, которая выдает ровно одно предсказание на каждой итерации. В методе IterDet не налагается ограничений на число объектов, найденных на одной итерации, и за счет этого IterDet является более вычислительно эффективным. В работе [16] для хранения информации об уже найденных объектах также применяется модуль долгой краткосрочной памяти (LSTM), в то время как в IterDet найденные объекты маскируются наложением двумерных масок специального вида. Такой способ имеет следующие преимущества над долгой краткосрочной памятью. Во-первых, это позволяет гарантировать, что все объекты, найденные на предыдущих итерациях, будут учтены в итоговом предсказании. Во-вторых, появляется возможность учитывать историю предсказаний на более глубоких уровнях нейросетевой модели. Кроме этого, модификация IterDet может быть легко применена к любой одно- или двухстадийной нейросетевой модели обнаружения объектов, что также является конкурентным преимуществом. Так, недавно представленный способ модификации для итеративного предсказания PS-RCNN [15] основан на схожей идее обнаружения сначала объектов в простых случаях, а затем в сложных. Однако он предполагает внесение значительных изменений в базовую нейросетевую модель обнаружения объектов. При этом возможности его применения ограничены, так как он предназначен только для двухстадийных моделей.

Экспериментальная проверка IterDet для одностадийной (RetinaNet [4]) и двухстадийной (Faster RCNN [3]) нейросетевых моделей обнаружения объектов была проведена на четырех различных наборах данных: Adaptis Toy V1, Toy V2 [17], CrowdHuman [12] и WiderPerson [18].

1. Обзор публикаций по теме

В настоящем разделе перечислены основные виды современных нейросетевых моделей обнаружения объектов в соответствии с общепринятой классификацией. Кратко описаны ключевые особенности моделей различных типов, основные сценарии их применения, возможности и ограничения моделей. Также описаны стандартные способы решения проблемы обнаружения сильно пересекающихся объектов: модификации алгоритма подавления немаксимумов и модификации, вносимые непосредственно в архитектуру нейросетевых моделей.

1.1. Стандартные методы обнаружения объектов

Все современные нейросетевые модели обнаружения объектов можно разделить на одно- и двухстадийные.

В двухстадийных моделях [2, 3] подзадачи предварительного отбора предсказаний-кандидатов и вычисления рамок объектов разделены, и для решения каждой подзадачи применяется отдельная нейросетевая модель. Первая модель выдает разреженное множество предсказаний-кандидатов, а вторая определяет, относится ли объект к одному из заданных классов или принадлежит к фону.

В одностадийных моделях применяется плотное сэмплирование позиций объектов с разными масштабами и отношениями сторон рамки объекта [1, 4]. Основным преимуществом одностадийных моделей обнаружения объектов является значительно меньшее количество вычислений, необходимое для получения предсказаний, и соответственно, более высокая скорость получения предсказаний. В ряде последних публикаций представлены одностадийные модели [5], не использующие фиксированные наборы гипотез о возможных позициях объектов. Подобные модели выигрывают в скорости у двухстадийных моделей, практически не уступая им в точности. Предложенный в настоящей статье способ модификации моделей для итеративного предсказания применим и к одно-, и к двухстадийным моделям.

Все упомянутые нейросетевые модели обнаружения объектов формулируют задачу обнаружения объектов как задачу классификации, в которой вероятности принадлежности объекта к каждому из классов вычисляются на некотором множестве положений объектов. При этом вероятности вычисляются независимо для каждого положения, в то время как между ними существует взаимосвязь, которая не учитывается в рамках описанных моделей. Предложенный автором итеративный метод позволяет учитывать множество ранее обнаруженных объектов, создавая таким образом контекст для разрешения неоднозначностей между перекрывающимися объектами.

1.2. Модификации метода подавления немаксимумов

На последнем этапе работы всех современных моделей обнаружения объектов применяется алгоритм подавления немаксимумов NMS. Стандартный алгоритм NMS "жадно" выбирает объекты с наибольшей вероятностью и отбрасывает их соседей с меньшими вероятностями. Варьируя уровень подавления немаксимумов, можно увеличивать либо точность, либо полноту решения задачи обнаружения объектов.

Изображения с большим числом пересекающихся объектов являются наиболее сложными для стандартного NMS. В нескольких недавно опубликованных статьях рассматриваются модификации стандартного метода подавления немаксимумов. В работе [6] NMS формулируется как задача кластеризации. В работе [8] рассматривается модификация NMS, в которой рамкам объектов, пересекающим рамки ранее

обнаруженных объектов, приписываются меньшие вероятности. В методе Soft NMS [7] вероятности предсказанных рамок объектов уменьшаются в зависимости от площади их пересечения с рамками ранее обнаруженных объектов. В методе Fitness NMS [9] эта идея получила дальнейшее развитие. В других статьях предлагается внести изменения и в архитектуру нейросетевой модели. Так, метод Adaptive NMS [10] подразумевает обучение дополнительной нейросетевой модели для оценки числа объектов в каждом пикселе изображения. Предсказанная этой моделью плотность распределения объектов на изображении затем учитывается в алгоритме подавления немаксимумов. Метод R^2NMS [11] позволяет достичь увеличения точности в случае, если для каждого объекта размечены две рамки — одна, включающая объект полностью, и вторая, ограничивающая только видимую часть объекта.

Все рассмотренные подходы оперируют полным множеством предсказаний. Предложенный в настоящей статье способ модификации нейросетевой модели обнаружения объектов позволяет получать предсказания итеративно, когда на каждой итерации обнаруживается часть полного множества объектов. Это, в частности, позволяет пропускать сложные объекты на первой итерации и затем возвращаться к ним на последующих итерациях. Таким образом, отсутствие ограничений по полноте предсказаний на каждой итерации позволяет повысить итоговую точность.

1.3. Альтернативные нейросетевые модели для большого числа объектов

За последние несколько лет был предложен ряд нейросетевых моделей обнаружения объектов на изображениях с большим числом сильно пересекающихся объектов. В работе [16] для последовательного обнаружения объектов используется рекуррентный модуль долгой краткосрочной памяти. Авторы также предлагают использовать венгерский алгоритм для вычисления функции потерь между истинным и предсказанным множествами ограничивающих рамок. Модель долгой краткосрочной памяти также используется в работе [19] для последовательного уточнения гипотез в двухстадийной нейросетевой модели обнаружения объектов. Однако в этой работе также применяется алгоритм подавления немаксимумов, что ограничивает применимость данного метода для большого числа пересекающихся объектов. В работе [13] описан оригинальный нейросетевой модуль, моделирующий отношения между найденными объектами, который учитывает их визуальные признаки и пространственную структуру. В другой публикации [14] предлагается использовать дополнительный нейросетевой слой, оценивающий меру Жаккара, которая затем используется в EM-алгоритме для разрешения неоднозначности перекрывающихся объектов. Модификация PS-RCNN [15] может быть применена к двухстадийной модели обнаружения объектов. На первом шаге находится слабо перекрывающееся подмножество объектов, которое затем маскируется. С учетом этой информации дополнительная модель обнаружения объектов находит сильно перекрывающиеся объекты.

В сравнении с описанными выше методами, IterDet легко применить к произвольной нейросетевой модели обнаружения объектов.

2. Описание метода

В настоящем разделе приведено математическое описание задачи обнаружения объектов и изложен алгоритм применения итеративного метода, а также описаны изменения процедуры обучения нейросетевой модели обнаружения объектов.

Типичная модель обнаружения объектов D трансформируется в алгоритм, отображающий цифровое трехканальное изображение $I \in R^{w \times h \times 3}$ во множество рамок объектов $B = \{(x_k, y_k, w_k, h_k)\}_{k=1}^{n}$, соответствующих объектам, содержащимся на этом изображении. Каждая рамка задается координатами ее левого верхнего угла (x, y), шириной w и высотой h.

Определим историю $H \in Z^{w \times h}$ для набора рамок B, как изображение такого же размера как I, в котором в каждом пикселе содержится число рамок, покрывающих этот пиксель:

$$H_{xy} = \sum_{k=1}^{|B|} 1_{x_k \leqslant x \leqslant x_k + w_k,\ y_k \leqslant y \leqslant y_k + h_k}.$$

Определим модель обнаружения объектов D' как *учитывающую историю*, если она принимает на вход историю H помимо изображения I.

Теперь приведем описание итеративного метода $IterDet(D')$ как алгоритма, отображающего изображение I во множество ограничивающих рамок B в несколько итераций. На первой итерации $t = 1$ история H_1 является пустой и D' отображает I и H_1 во множество ограничивающих рамок B_1. Затем по B_1 строится история H_2, которая, в свою очередь, совместно с I на итерации $t = 2$ отображается D' во множество B_2. Этот процесс останавливается при достижении ограничения на число итераций или на итерации m, такой что $|B_m| = 0$. Итоговым предсказанием $IterDet(D')$ является $B = \bigcup_{t=1}^{m} B_t$.

Представленное выше описание итеративного метода оставляет без ответа два ключевых вопроса: как модифицировать стандартную модель обнаружения объектов D в учитывающую историю D'; как обучить D' предсказывать непересекающиеся подмножества ограничивающих рамок B_t на каждой итерации t. Оба вопроса рассматриваются ниже.

Вначале все современные нейросетевые модели обнаружения объектов извлекают из изображений признаки с помощью предобученной базовой нейронной сети, например, VGG [20], ResNet [21] и др. Затем эти признаки трансформируются промежуточной нейронной сетью, такой как RPN (*Region Proposal Network*), FPN (*Feature Pyramid Network*) и т. д. Наконец, финальные слои нейросетевой модели выдают набор ограничивающих рамок с соответствующими

вероятностями, которые затем фильтруются алгоритмом подавления немаксимумов. При применении модификации IterDet в описанную выше схему предлагается внести минимальные изменения, а именно, объединять признаки из изображения и информацию из истории на самых ранних слоях нейросетевой модели.

Предлагаемая архитектура нейросетевой модели обнаружения объектов, учитывающей историю, оказывается простой и в то же время эффективной. История пропускается через один сверточный слой и затем складывается с выходом первого сверточного слоя предобученной базовой нейросетевой модели. Такое изменение архитектуры может быть применено к любой базовой сети без дальнейшего подбора гиперпараметров. Во всех экспериментах, описанных в настоящей статье, в качестве предобученной базовой сети используется ResNet.

Приведем более подробное описание архитектуры для этого случая. В сетях архитектуры ResNet поступающее на вход изображение сначала трансформируется сверточным слоем с 64 фильтрами размера 7 и шагом 2, слоем нормализации и ReLU-активацией. Поскольку признаки, полученные из истории, складываются с признаками изображения после этих слоев, то параметры сверточного слоя истории, задаются такими же, а именно, этот слой задается 64 фильтрами размера 3 и шагом 2. Здесь размер фильтра не важен, поэтому 7 заменено на более часто встречающееся значение 3.

Предлагаемая архитектура итеративного метода обнаружения объектов приведена на рис. 2 (см. третью сторону обложки). Стандартная архитектура нейросетевой модели обнаружения объектов показана синим цветом. Добавленный блок для отображения истории отмечен зеленым. Из четырех перекрывающихся объектов на изображении два находятся в истории, куда они были случайно выбраны на этапе обучения или найдены на предыдущих итерациях при тестировании. Оставшиеся два объекта предсказываются моделью.

Для предсказания непересекающихся множеств объектов на каждой итерации при тестировании шаг обучения также должен быть модифицирован. Для каждого изображения множество истинных рамок объектов $\hat{B}$ случайным образом разбивается на два подмножества B_{old} и B_{new}, таких что $B_{old} \cup B_{new} = \hat{B}$ и $B_{old} \cap B_{new} = \varnothing$. При этом B_{old} отображаются в историю H, D' обучается предсказывать только ограничивающие рамки из B_{new}, которых нет в истории. Для этого функция потерь D' оптимизируется методом обратного распространения ошибки между предсказываемым множеством B и истинным B_{new}.

В ряде публикаций [16, 22] описаны методы, предсказывающие один объект на каждой итерации. Отметим, что IterDet также позволяет предсказывать один объект на итерацию, выбирая ограничивающую рамку с наибольшей вероятностью. Однако на практике такой подход оказывается неэффективным, так как время выполнения алгоритма пропорционально числу объектов на нем. Эксперименты, описанные в разд. 3, демонстрируют, что двух итераций оказывается достаточно для достижения лучшей точности. Дальнейшее увеличение числа итераций приводит к увеличению полноты и уменьшению точности, ухудшая в итоге целевые метрики.

3. Эксперименты

В первой части настоящего раздела изложены условия экспериментов: описаны используемые наборы данных, а также процедура обучения нейросетевых моделей. Далее приведены результаты экспериментов в виде сводных таблиц и проведен анализ полученных результатов.

3.1. Наборы данных и детали реализации

Экспериментальная проверка IterDet была проведена на четырех наборах данных: AdaptIS Toy V1 и Toy V2 [17], CrowdHuman [12] и WiderPerson [18].

Синтетические наборы данных AdaptIS Toy V1 и Toy V2 были изначально представлены в работе [17] для задачи сегментации объектов. Каждое изображение в этих наборах данных содержит до 50 сильно перекрывающихся объектов. Более подробная статистика числа объектов и пар объектов с различными уровнями пересечения в среднем на одном изображении приведена в табл. 1.

Обучающая и валидационная выборки Toy V1 содержат 10 000 и 2000 изображений размера 96×96 пикселей соответственно. Набор Toy V2 разбит на три выборки: обучающую, валидационную и тестовую с 25 000, 1000 и 1000 изображениями размера 128×129

Таблица 1

Сравнение наборов данных по среднему числу объектов и пар объектов с различными уровнями пересечения на одном изображении

Набор данных	Среднее число объектов на изображении	Среднее число пар пересекающихся объектов на изображении			
		IoU > 0,3	IoU > 0,4	IoU > 0,5	IoU > 0,6
Toy V1	14,88	3,67	1,95	0,95	0,38
Toy V2	31,25	7,12	3,22	1,25	0,45
CrowdHuman	22,64	9,02	4,89	2,40	1,01
WiderPerson	29,51	9,21	4,78	2,15	0,81

пикселей соответственно. Обучение проводилось на смеси обучающей и валидационной выборок. В качестве целевой метрики на наборах данных Toy V1, Toy V2 была выбрана средняя точность (*average precision*, AP).

Набор данных CrowdHuman содержит фотографии массовых скоплений людей. Это самый сложный набор данных согласно сравнительному анализу, приведенному в работе [12], где сложность определялась через среднее число объектов на одном изображении и количество пересекающихся пар объектов с отношением площадей пересечения и объединения (*intersection over union*, IoU) > 0,5 (см. табл. 1). Обучающая, валидационная и тестовая подвыборки этого набора содержат соответственно 15 000, 4370 и 5000 изображений. Эталонная разметка для тестовой подвыборки не была опубликована, поэтому тестирование и сравнение с другими нейросетевыми моделями обнаружения объектов проведено на валидационной подвыборке. На каждом изображении набора данных CrowdHuman присутствуют в среднем 23 человека, для каждого размечены три рамки: одна содержит фигуру человека целиком, вторая — видимую часть фигуры человека, третья — только голову человека. Самым распространенным видом разметки является разметка человека полностью. Она же представляет наибольшую сложность для моделей обнаружения объектов, так как число пар пересекающихся объектов больше, а рамки объектов могут выходить за края изображения. По этой причине именно этот вид разметки был выбран для проведения экспериментальной проверки в первую очередь. Кроме этого, проведены эксперименты по обнаружению видимой части тела человека.

Эффективность моделей обнаружения объектов оценивается с помощью метрик, описанных в работе [12]. Помимо стандартных метрик, т. е. полноты и средней точности AP, используется метрика mMR. Значение этой метрики вычисляется как логарифм отношения числа ложноотрицательных срабатываний к числу ложноположительных, усредненного по различным значениям числа ложноположительных объектов на изображении от 10^{-2} до 10^{0}.

Аналогично CrowdHuman, WiderPerson [18] содержит данные из различных источников. В его разметке присутствуют объекты пяти классов: пешеходы, велосипедисты, люди с сильными перекрытиями, толпы людей и игнорируемые области. Так же, как и в методе, описанном в статье [15], последние четыре класса были объединены в один. Обучающая, валидационная и тестовая подвыборки этого набора содержат 8000, 1000 и 4382 изображения соответственно. Как и для набора CrowdHuman, разметка тестовой подвыборки не была опубликована, поэтому все модели тестируются на валидационной подвыборке.

Программный код для проведения всех экспериментов по обнаружению объектов реализован на языке Python на основе библиотеки MMDetection [23], которая, в свою очередь, использует инструментальные средства библиотеки глубокого обучения PyTorch [24]. Библиотека MMDetection содержит имплементации более десятка одно- и двухстадийных нейросетевых моделей обнаружения объектов, имеет модульный дизайн, позволяющий легко интегрировать IterDet. Для экспериментов в качестве базовой модели приняты RetinaNet и Faster RCNN. В обоих моделях в качестве предобученной базовой сети используется ResNet-50. Большая часть гиперпараметров инициализирована значениями по умолчанию из MMDetection. Для обучения были задействованы восемь видеокарт, в процессе обучения на каждую подавалось по два изображения единовременно.

Большая часть деталей имплементации нейросетевых моделей и процедуры их обучения одинаковы для всех наборов данных. В нейросетевые модели обнаружения объектов был внесен ряд описанных далее изменений. Во-первых, был добавлен нейросетевой слой, выполняющий нормализацию выходов после каждого сверточного слоя в FPN. Для части моделей это привело к повышению точности. Во-вторых, первый блок ResNet был сделан обучаемым, так как перед ним добавлен обучаемый слой, агрегирующий информацию из истории. Во всех экспериментах с IterDet использован метод оптимизации Adam с начальной скоростью обучения 0,0001. Для экспериментов без IterDet использован метод оптимизации SGD с моментом, равным 0,9, параметром регуляризации весов 0,0001 и начальной скоростью обучения 0,02, после чего скорость обучения уменьшалась в 10 раз после 16-й и 22-й эпох обучения (прохода всего набора данных через нейросетевую модель). В общей сложности в обоих случаях обучение проводилось 24 эпохи.

Изображения набора CrowdHuman предварительно обрабатываются, как это было предложено в работе [12]. На этапе тестирования изображения масштабируются, чтобы меньшая по длине сторона была не более 800 пикселей, а длина большей стороны не превышала 1400 пикселей. На этапе тестирования не используются аугментации изображений. На этапе обучения используются две аугментации: случайное горизонтальное отражение и случайное изменение размера в пределах 25 %. Учет информации об игнорируемых регионах на изображении также улучшает итоговую точность на CrowdHuman.

В экспериментах на наборе CrowdHuman с разметкой человека полностью используются пять якорей с отношениями сторон 1, 1,5, 2, 2,5, 3, как было предложено в работах [10, 12, 15]. Изображения наборов AdaptIS Toy V1 и Toy V2 масштабируются до размера 384×384 пикселя как и в работе [17]. Протокол экспериментов на наборе WiderPerson полностью соответствует протоколу для набора CrowdHuman.

3.2. Результаты

В табл. 2 приведены результаты экспериментов на наборах данных Toy V1 и Toy V2.

В таблицах полужирным шрифтом выделены лучшие значения каждой из метрик для каждой нейросетевой модели обнаружения объектов. Для обеих базовых моделей RetinaNet и Faster RCNN и для обоих наборов данных Toy V1 и Toy V2 IterDet показывает существенный прирост средней точности (4 %) на

Таблица 2

Результаты экспериментов на наборах данных Toy V1 и Toy V2

Нейросетевая модель обнаружения объектов	Метод	Полнота, %		Точность, %	
		Toy V1	Toy V2	Toy V1	Toy V2
RetinaNet	Базовый	95,46	96,27	94,46	95,62
	IterDet, одна итерация	95,21	96,27	95,31	94,17
	IterDet, две итерации	**99,56**	**99,35**	**97,71**	**97,27**
Faster RCNN	Базовый	94,05	94,88	93,96	94,81
	IterDet, одна итерация	94,34	94,97	94,27	94,89
	IterDet, две итерации	**99,60**	**99,29**	**99,25**	**99,00**

второй итерации в сравнении с базовым методом. Финальная средняя точность для Faster RCNN составляет 99 %.

Результаты на наборе CrowdHuman с двумя разметками — фигуры полностью и видимой части тела человека, — приводятся в табл. 3 и 4.

Сравнение приведено со всеми известными автору методами, не использующими дополнительных данных. В таблицах пропущены значения некоторых метрик, которые не были приведены авторами соответствующих работ. Основным результатом представленных исследований является значительное улучшение всех трех метрик на наборе CrowdHuman с разметкой человека целиком, см. последние две строки табл. 3. Так, IterDet улучшает полноту более чем на 5,5 %, среднюю точность — на 3,1 % и mMR на 1 % по сравнению с базовым методом. Улучшение значений метрик заметно и в сравнении с предыдущими лучшими методами Adaptive NMS и PS-RCNN. По метрике mMR на этом наборе данных IterDet позволяет повысить качество предсказаний во всех сценариях: для одно- и двухстадийных моделей обнаружения объектов и для обоих типов разметки. В случае с RetinaNet разница в метриках составляет 6 % в пользу IterDet по сравнению с моделью без модификации. По метрике mMR улучшение во всех сценариях достигается на первой итерации. Это свидетельствует о том, что учет предыдущих предсказаний позволяет лучше обучать нейросетевую модель обнаружения объектов. Несмотря на небольшое ухудшение mMR на второй итерации, рост средней точности во всех случаях остается значительным. Так, относительно лучшего из существующих результатов с использованием RetinaNet прирост средней точности превышает 3,9 % для обоих типов разметки.

Результаты на наборе данных WiderPerson приведены в табл. 5.

Результаты базового метода приводятся по статье [18]. Эти результаты были получены авторами базового метода в эксперименте по обнаружению объектов размером не менее 20 пикселей. Так же, как и в работе [15], итеративный метод тестируется

Таблица 3

Результаты экспериментов на наборе данных CrowdHuman с разметкой фигуры полностью

Нейросетевая модель обнаружения объектов	Метод	Полнота, %	Точность, %	mMR
RetinaNet	Базовый [12]	**93,80**	80,83	63,66
	IterDet, одна итерация	79,68	76,78	**53,03**
	IterDet, две итерации	91,49	**84,77**	56,21
Faster RCNN	Базовый [12]	90,24	84,95	50,49
	Soft NMS [7, 10]	91,73	83,92	51,97
	Adaptive NMS [10]	91,27	84,71	49,73
	Rep. Loss [15, 25]	90,74	85,71	—
	PS-RCNN [15]	93,77	86,05	—
	IterDet, одна итерация	88,94	84,43	**49,12**
	IterDet, две итерации	**95,80**	**88,08**	49,40

Таблица 4

Результаты экспериментов на наборе данных CrowdHuman с разметкой видимой части тела человека

Нейросетевая модель обнаружения объектов	Метод	Полнота, %	Точность, %	mMR
RetinaNet	Базовый [12]	**90,96**	77,19	65,47
	Feature NMS [26]	—	68,65	75,35
	IterDet, одна итерация	86,91	81,24	**58,78**
	IterDet, две итерации	89,63	**82,32**	59,19
Faster RCNN	Базовый [12]	91,51	**85,60**	55,94
	IterDet, одна итерация	87,59	83,28	**55,54**
	IterDet, две итерации	**91,63**	85,33	55,61

Таблица 5

Результаты экспериментов на наборе данных WiderPerson

Нейросетевая модель обнаружения объектов	Метод	Полнота, %	Точность, %	mMR
RetinaNet	Базовый [18]	—	—	48,32
	IterDet, одна итерация	90,38	87,17	**43,23**
	IterDet, две итерации	**95,35**	**90,23**	43,88
FasterRCNN	Базовый [18]	—	—	46,06
	Базовый [15]	93,60	88,89	—
	IterDet, одна итерация	94,71	89,96	—
	IterDet, две итерации	92,67	89,49	**40,35**

в более сложных условиях, а именно, без ограничения объектов по размеру. Согласно значениям всех трех метрик, итеративный метод превосходит все существующие нейросетевые модели обнаружения объектов.

В некоторых публикациях, ссылки на которые даны в работе [15], для оценки эффективности использования нейросетевых моделей обнаружения объектов применяют обе метрики — средняя точность и mMR, поскольку их оптимальные значения могут достигаться не одновременно. В настоящей статье для наборов CrowdHuman и WiderPerson были вычислены значения обеих метрик. Из данных табл. 3, 4 и 5 следует, что метрика mMR достигает минимального значения при одной итерации. На рис. 3 (см. третью сторону обложки) сравниваются значения средней точности при разном числе итераций для Faster RCNN-IterDet.

Для всех пяти наборов данных наблюдается сильный прирост качества между одной и двумя итерациями. Дальнейшее увеличение числа итераций не приводит к росту средней точности. Небольшое падение средней точности может быть компенсировано дополнительным порогом на шаге подавления немаксимумов, увеличивающимся с номером итерации. Однако детальное исследование наблюдаемого эффекта выходит за рамки настоящей статьи, поскольку использование IterDet позволяет превзойти существующие методы в точности уже за две итерации. При этом время выполнения итеративной схемы пропорционально числу итераций, и увеличение длительности получения предсказаний более чем в 3 раза при использовании трех и более итераций ограничивает возможности практического применения метода.

На рис. 4 (см. четвертую сторону обложки) показаны результаты обнаружения объектов с помощью IterDet на четырех наборах данных. Ограничивающие рамки объектов, найденных на первой и второй итерациях, обозначены соответственно зеленым и желтым цветом. Визуализированы все истинно положительные рамки с предсказанной вероятностью больше 0,1. На всех изображениях присутствуют сильно пересекающиеся объекты с IoU > 0,5, которые не могли быть обнаружены базовой моделью, в отличие от модели с модификацией IterDet с двумя итерациями

В дополнительном эксперименте исследуется вариант IterDet с дополнительным ограничением: за одну итерацию может быть обнаружен только один объект. Для этого алгоритм подавления немаксимумов был заменен на выбор объекта с наибольшей предсказанной вероятностью. В такой постановке процедура обучения не меняется, а во время тести-

рования модель останавливается, если на текущей итерации было обнаружено 0 объектов.

Заметим, что время, затраченное на получение предсказаний с помощью нейросетевой модели, пропорционально числу объектов на изображении, что не позволяет применять описанную модификацию на практике, тем не менее, результаты подобного эксперимента имеют теоретическое значение. Этот вариант IterDet также повышает точность обнаружения объектов относительно базовых моделей. Так, средняя скорость достигает 98,39 % на наборе Toy V2, что значительно превышает результат базового метода из табл. 2. Промежуточные шаги итеративного метода с ограничением числа объектов на итерацию изображены на рис. 5 (см. четвертую сторону обложки). Первый ряд содержит визуализации масок ранее найденных объектов. В изображениях второго ряда объекты, найденные на каждой итерации, обведены рамкой. Справа приведен итоговый результат: за 16 итераций были успешно найдены 16 объектов.

Заключение

Описан способ модификации нейросетевых моделей обнаружения объектов IterDet, позволяющий повысить точность предсказаний на изображениях с большим числом пересекающихся объектов. Предложенным способом, получившим название IterDet, были модифицированы одно- и двухстадийные нейросетевые модели обнаружения объектов. Эксперименты на наборах данных AdaptIS Toy V1 и Toy V2 с большим числом пересекающихся объектов продемонстрировали, что применение IterDet позволяет достичь практически идеальной точности. На фотографиях массовых скоплений людей из наборов данных CrowdHuman и WiderPerson метод IterDet повышает точность обнаружения людей относительно базовых нейросетевых моделей, позволяя таким образом получить лучшие на момент написания настоящей статьи результаты среди всех других методов обнаружения объектов.

Список литературы

1. **Liu W., Anguelov D., Erhan D.** et al. Ssd: Single shot multibox detector // European conference on computer vision. — Springer, Cham. — 2016. — P. 21—37.

2. **Girshick R.** Fast r-cnn // Proceedings of the IEEE International Conference on Computer Vision. — 2015. — P. 1440—1448.

3. **Ren S., He K., Girshick R.** et al. Faster r-cnn: Towards real-time object detection with region proposal networks // Advances in neural information processing systems. — 2015. — P. 91—99.

4. **Lin T., Goyal P., Girshick R.** et al. Focal loss for dense object detection // Proceedings of the IEEE International Conference on Computer Vision. — 2017. — P. 2980—2988.

5. **Tian Z., Shen C., Chen H.** et al. Fcos: Fully convolutional one-stage object detection // Proceedings of the IEEE International Conference on Computer Vision. — 2019. — P. 9627—9636.

6. **Rothe R., Guillaumin M., Van Gool L.** Non-maximum suppression for object detection by passing messages between windows // Asian conference on computer vision. — Springer, Cham. — 2014. — P. 290—306.

7. **Bodla N., Singh B., Chellappa R.** et al. Soft-NMS—improving object detection with one line of code // Proceedings of the IEEE international conference on computer vision. — 2017. — P. 5561—5569.

8. **Hosang J., Benenson R., Schiele B.** Learning non-maximum suppression // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — P. 4507—4515.

9. **Tychsen-Smith L., Petersson L.** Improving object localization with fitness nms and bounded iou loss // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2018. — P. 6877—6885.

10. **Liu S., Huang D., Wang Y.** Adaptive nms: Refining pedestrian detection in a crowd // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2019. — P. 6459—6468.

11. **Huang X., Ge Z., Jie Z.** et al. NMS by Representative Region: Towards Crowded Pedestrian Detection by Proposal Pairing // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2020. — P. 10750—10759.

12. **Shao S., Zhao Z., Li B.** et al. Crowdhuman: A benchmark for detecting human in a crowd // arXiv preprint arXiv:1805.00123. — 2018.

13. **Hu H., Gu J., Zhang Z.** et al. Relation networks for object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recogntion. — 2018. — P. 3588—3597.

14. **Goldman E., Herzig R., Eisenschtat A.** et al. Precise detection in densely packed scenes // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2019. — P. 5227—5236.

15. **Ge Z., Jie Z, Huang X.** et al. Ps-rcnn: Detecting secondary human instances in a crowd via primary object suppression // 2020 IEEE International Conference on Multimedia and Expo (ICME). — IEEE, 2020. — P. 1—6.

16. **Stewart R., Andriluka M., Ng A.** End-to-end people detection in crowded scenes // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2016. — P. 2325—2333.

17. **Sofiiuk K., Barinova O., Konushin A.** Adaptis: Adaptive instance selection network // Proceedings of the IEEE International Conference on Computer Vision. — 2019. — P. 7355—7363.

18. **Zhang S., Xie Y., Wan J.** et al. Widerperson: A diverse dataset for dense pedestrian detection in the wild // IEEE Transactions on Multimedia. — 2019. — Vol. 22, No. 2. — P. 380—393.

19. **Gong J., Zhao Z., Li N.** Improving Multi-stage Object Detection via Iterative Proposal Refinement // BMVC. — 2019. — P. 223.

20. **Simonyan K., Zisserman A.** Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556. — 2014.

21. **He K., Zhang X., Ren S.** et al. Deep residual learning for image recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2016. — P. 770—778.

22. **Barinova O., Lempitsky V., Kholi P.** On detection of multiple object instances using hough transforms // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2012. — Vol. 34, No. 9. — P. 1773—1784.

23. **Chen K., Wang J., Pang J.** et al. Mmdetection: Open mmlab detection toolbox and benchmark // arXiv preprint arXiv:1906.07155. — 2019.

24. **Paszke A., Gross S., Massa F.** et al. Pytorch: An imperative style, high-performance deep learning library // Advances in neural information processing systems. — 2019. — P. 8026—8037.

25. **Wang X., Xiao T., Jiang Y.** et al. Repulsion loss: Detecting pedestrians in a crowd // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2018. — P. 7774—7783.

26. **Salscheider N.** FeatureNMS: Non-Maximum Suppression by Learning Feature Embeddings // arXiv preprint arXiv:2002.07662. — 2020.

Iterative Scheme for Object Detection in Crowded Environments

D. D. Rukhovich, d.rukhovich@samsung.com, Samsung AI Center Moscow, Moscow, 127018, Russian Federation

Corresponding author:

Rukhovich Danila D., Engineer, Samsung AI Center Moscow, Moscow, 127018, Russian Federation
E-mail: d.rukhovich@samsung.com

Received on September 14, 2020
Accepted on October 21, 2020

Deep learning-based detectors usually produce a redundant set of object bounding boxes including many duplicate detections of the same object. These boxes are then filtered using non-maximum suppression (NMS) in order to select exactly one bounding box per object of interest. This greedy scheme is simple and provides sufficient accuracy for isolated objects but often fails in crowded environments, since one needs to both preserve boxes for different objects and suppress duplicate detections. In this work we develop an alternative iterative scheme, where a new subset of objects is detected at each iteration. Detected boxes from the previous iterations are passed to the network at the following iterations to ensure that the same object would not be detected twice. This iterative scheme can be applied to both one-stage and two-stage object detectors with just minor modifications of the training and inference procedures. We perform extensive experiments with two different baseline detectors on four datasets and show significant improvement over the baseline, leading to state-of-the-art performance on CrowdHuman and WiderPerson datasets.

Keywords: *deep learning, object detection, non-maximum suppression*

For citation

Rukhovich D. D. Iterative Scheme for Object Detection in Crowded Environments, *Programmnaya Ingeneria,* 2021, vol. 12, no. 1, pp. 31—39

DOI: 10.17587/prin.12.31-39

References

1. **Liu W., Anguelov D., Erhan D.** et al. Ssd: Single shot multibox detector, *European conference on computer vision*, Springer, Cham, 2016, pp. 21—37.
2. **Girshick R.** Fast r-cnn, *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440—1448.
3. **Ren S., He K., Girshick R.** et al. Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 2015, pp. 91—99.
4. **Lin T., Goyal P., Girshick R.** et al. Focal loss for dense object detection, *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980—2988.
5. **Tian Z., Shen C., Chen H.** et al. Fcos: Fully convolutional one-stage object detection, *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627—9636.
6. **Rothe R., Guillaumin M., Van Gool L**. Non-maximum suppression for object detection by passing messages between windows, *Asian conference on computer vision*, Springer, Cham, 2014, pp. 290—306.
7. **Bodla N., Singh B., Chellappa R.** et al. Soft-NMS—improving object detection with one line of code, *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561—5569.
8. **Hosang J., Benenson R., Schiele B.** Learning non-maximum suppression, *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2017, pp. 4507—4515.
9. **Tychsen-Smith L., Petersson L.** Improving object localization with fitness nms and bounded iou loss, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6877—6885.
10. **Liu S., Huang D., Wang Y.** Adaptive nms: Refining pedestrian detection in a crowd, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6459—6468.
11. **Huang X., Ge Z., Jie Z.** et al. NMS by Representative Region: Towards Crowded Pedestrian Detection by Proposal Pairing, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2020, pp. 10750—10759.
12. **Shao S., Zhao Z., Li B.** et al. Crowdhuman: A benchmark for detecting human in a crowd, arXiv preprint arXiv:1805.00123, 2018.
13. **Hu H., Gu J., Zhang Z.** et al. Relation networks for object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3588—3597.
14. **Goldman E., Herzig R., Eisenschtat A.** et al. Precise detection in densely packed scenes, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2019, pp. 5227—5236.
15. **Ge Z., Jie Z, Huang X.** et al. Ps-rcnn: Detecting secondary human instances in a crowd via primary object suppression, *2020 IEEE International Conference on Multimedia and Expo (ICME),* IEEE, 2020, pp. 1—6.
16. **Stewart R., Andriluka M., Ng A.** End-to-end people detection in crowded scenes, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2016, pp. 2325—2333.
17. **Sofiiuk K., Barinova O., Konushin A.** Adaptis: Adaptive instance selection network, *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7355—7363.
18. **Zhang S., Xie Y., Wan J.** et al. Widerperson: A diverse dataset for dense pedestrian detection in the wild, *IEEE Transactions on Multimedia*, 2019, vol. 22, no. 2, pp. 380—393.
19. **Gong J., Zhao Z., Li N.** Improving Multi-stage Object Detection via Iterative Proposal Refinement, *BMVC,* 2019, pp. 223.
20. **Simonyan K., Zisserman A.** Very deep convolutional networks for large-scale image recognition,arXiv preprint arXiv:1409.1556, 2014.
21. **He K., Zhang X., Ren S.** et al. Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770—778.
22. **Barinova O., Lempitsky V., Kholi P.** On detection of multiple object instances using hough transforms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, vol. 34, no. 9, pp. 1773—1784.
23. **Chen K., Wang J., Pang J.** et al. Mmdetection: Open mmlab detection toolbox and benchmark, arXiv preprint arXiv:1906.07155, 2019.
24. **Paszke A., Gross S., Massa F.** et al. Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems,* 2019, pp. 8026—8037.
25. **Wang X., Xiao T., Jiang Y.** et al. Repulsion loss: Detecting pedestrians in a crowd, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7774—7783.
26. **Salscheider N.** FeatureNMS: Non-Maximum Suppression by Learning Feature Embeddings, arXiv preprint arXiv:2002.07662, 2020.

УДК 004.825 DOI: 10.17587/prin.12.40-47

С. Д. Махортов, д-р физ.-мат. наук, зав. кафедрой, msd_exp@outlook.com,
Воронежский государственный университет

О разрешимости и числе решений продукционно-логических уравнений в нечеткой LP-структуре

Разработанная автором теория LP-структур предназначена для формализации методов управления знаниями в интеллектуальных системах продукционного типа. Одно из ее направлений — методы релевантного обратного вывода, снижающие число обращений к внешним источникам информации. Они основаны на решении продукционно-логических уравнений. В статье рассматривается расширенный класс таких уравнений для алгебраической модели, выразительные возможности которой охватывают нечеткие интеллектуальные системы. С учетом доказанных ранее свойств уравнений исследованы вопросы о разрешимости и числе решений. Нахождение решения продукционно-логического уравнения соответствует обратному нечеткому логическому выводу. Представленные результаты служат теоретической основой для дальнейших продвижений в области его оптимизации. Обсуждены идеи практического применения уравнений для ускорения нечеткого обратного вывода.

Ключевые слова: *нечеткая продукционная система, релевантный обратный вывод, алгебраическая модель, нечеткая LP-структура, продукционно-логическое уравнение*

Введение

Для построения и исследования формальных моделей интеллектуальных информационных систем оказываются полезными алгебраические методы [1, 2]. Одно из актуальных направлений здесь — широко распространенные в информатике логические системы продукционного типа [3—5].

В последние годы автором и его последователями разрабатывается алгебраическая теория LP-структур (*lattice production structures*) [6]. Она предназначена для формализации и решения ряда задач управления знаниями в продукционных системах. Был также введен и исследован метод релевантного обратного вывода (LP-вывод) [7], значительно снижающий число обращений к внешним источникам информации по сравнению с классическим выводом. Впоследствии теория была обобщена для ускорения логического вывода в распределенных продукционных системах [8].

В то же время современные интеллектуальные системы характеризуются нечеткостью знаний и рассуждений [9]. Поэтому возникает потребность в распространении теории LP-структур на нечеткие продукционные системы. Начало этим исследованиям положено в работах [10, 11]. Введены некоторые понятия, придающие LP-структурам нечеткость, а также установлены отдельные свойства нечеткого LP-вывода. В работах [12, 13] представлены результаты исследований, систематически обобщающие теорию LP-структур на случай нечетких баз знаний. Введена терминология FLP-структур с нечетким логическим отношением (*Fuzzy LP-структуры*), доказаны основные стандартные свойства.

Настоящая работа дополняет модель FLP-структур исследованием класса продукционно-логических уравнений. В результате создается теоретическая основа для формального исследования и ускорения обратного *нечеткого логического вывода*.

Статья состоит из следующих основных разделов. В разд. 1 введены базовые понятия и обозначения, сформулированы исходные математические результаты. В разд. 2 приведено определение класса продукционно-логических уравнений в нечеткой LP-структуре, сформулирована теорема, лежащая в основе их исследования. В разд. 3 перечислены свойства уравнений и описаны методы их упрощения, строгое обоснование которых запланировано для публикации в отдельной статье. В разд. 4 сформулированы и доказаны основные результаты настоящей работы — о существовании и способе нахождения решений для рассматриваемого класса уравнений. В разд. 5 обсуждены идеи практического применения нечетких продукционно-логических уравнений для ускорения нечеткого обратного вывода (релевантного FLP-вывода). В Заключении подведены итоги и обсуждены направления продолжения исследований.

1. Исходные понятия теории FLP-структур

В настоящем разделе используется терминология теории решеток, нечетких множеств и бинарных отношений. Она изложена, в частности, в работах [14, 15]. Напомним некоторые из этих понятий.

Нечеткое множество $A = (F, \mu_A)$ задается функцией принадлежности $\mu_A : F \to [0, 1]$ на ординарном ("четком") множестве F. Величина $\mu_A(a)$ называется степенью принадлежности a к F. Носителем S_A не-

четкого множества A (S_A = support A) называется четкое подмножество всех элементов $a \in F$, на которых $\mu_A(a) > 0$. Нечеткое множество конечно, если конечен его носитель.

Нечеткое бинарное отношение R на множестве F — это нечеткое множество упорядоченных пар элементов из F с функцией принадлежности $\mu_R : F \times F \to [0, 1]$. Нечеткое отношение R на множестве F называется *рефлексивным*, если для любого $a \in F$ справедливо $\mu_R(a, a) = 1$.

Для моделирования нечеткого логического вывода будем использовать композицию нечетких отношений в классической семантике — (max-min)-композицию. Отношение $R^2 = R \circ R$ определяется следующим образом:

$$\mu_{R^2}(a, c) = \max_b \left(\min\left(\mu_R(a, b), \mu_R(b, c)\right)\right),$$

где $a, b, c \in F$.

Нечеткое бинарное отношение R на множестве F *транзитивно*, если для любых $a, b, c \in F$ справедливо $\mu_R(a, c) \geqslant \min\left(\mu_R(a, b), \mu_R(b, c)\right)$. Существует замыкание произвольного нечеткого отношения относительно свойств рефлексивности и транзитивности. Обзор алгоритмов его построения представлен в работе [16].

Пусть дана атомно-порожденная решетка $\mathbb{F}$, представляющая множество всех конечных подмножеств универсума F. Чтобы подчеркнуть этот факт, будем вместо символов $\leqslant$, $\geqslant$, $\wedge$ и $\vee$ (принятых в общей теории решеток [14]) использовать знаки теоретико-множественных операций $\subseteq$, $\supseteq$, $\bigcap$ и $\bigcup$, а элементы решетки обозначать, как правило, большими буквами. Исключение составляют атомы, указываемые маленькими буквами.

На $\mathbb{F}$ вводится нечеткое бинарное отношение R, содержащее $\supseteq$, а также обладающее свойствами транзитивности и дистрибутивности. Последнее свойство имеет следующую семантику [12].

Определение 1.1. Нечеткое бинарное отношение $R = (\mathbb{F}, \mu_R)$ называется дистрибутивным, если для любых $A, B_1, B_2 \in \mathbb{F}$ справедливо $\mu_R(A, B_1 \cup B_2) \geqslant \min\left(\mu_R(A, B_1), \mu_R(A, B_2)\right)$.

Отношение с упомянутыми выше тремя свойствами будем называть продукционно-логическим (для краткости — просто логическим).

Определение 1.2. Под нечеткой LP-структурой (FLP-структурой) подразумевается алгебраическая система, представляющая собой решетку с заданным на ней нечетким продукционно-логическим отношением.

В работе [12] показано, что такая алгебраическая система может служить формальной моделью интеллектуальной системы продукционного типа с нечеткими правилами. Введенное на ней бинарное отношение моделирует множество нечетких продукций. Его свойства отражают возможности нечеткого логического вывода на решетке.

Для дальнейшего изложения потребуются некоторые связанные с FLP-структурами результаты, доказанные в работах [12, 13].

Исходное отношение R, моделирующее множество продукций, обычно не является логическим, но может быть рассмотрено его логическое замыкание. Замыкание моделирует все потенциальные логические выводы в продукционной системе.

Логическим замыканием $\bar{R}$ нечеткого бинарного отношения R называется наименьшее логическое отношение, содержащее R.

В работе [12] доказано существование логического замыкания и описана его структура. Этот факт позволил ввести понятие эквивалентных отношений, т. е. в приложениях — формально эквивалентных нечетких баз знаний.

Два нечетких отношения R, P на общей решетке называются (*логически*) *эквивалентными* $(R \sim P)$, если их логические замыкания совпадают. *Эквивалентным преобразованием* нечеткого отношения R называется такая модификация его функции принадлежности $(\mu_R \to \mu_P)$, что полученное в результате новое отношение P логически эквивалентно R.

Нечеткое отношение на атомно-порожденной решетке $\mathbb{F}$ называется *каноническим*, если его функция принадлежности положительна лишь на парах вида (A, a), где $A \in \mathbb{F}$, a — атом в $\mathbb{F}$. В работе [12] доказана теорема о существовании эквивалентного канонического отношения для произвольного R. В моделируемой продукционной системе каноническое отношение соответствует множеству правил так называемого *хорновского* типа.

В работе [13] исследовано содержание логических связей в нечеткой LP-структуре. Доказано, что логическое замыкание нечеткого отношения R совпадает с транзитивным замыканием отношения $\tilde{R} \supseteq R$, построенного по R в виде "дистрибутивного многообразия". Приведем формулировку этого результата, поскольку она потребуется в дальнейшем изложении.

Для нечеткого отношения R на решетке $\mathbb{F}$ вводится отношение $\tilde{R}$ (с функцией принадлежности $\mu_{\tilde{R}} = \tilde{\mu}_R$), построенное последовательным выполнением следующих действий (шагов).

1. Объединить R с отношением рефлексивности на решетке $\mathbb{F}$ и обозначить новое отношение R_1.

2. Расширить R_1 всевозможными парами (A, B), где $A = \bigcup A_t$, $B = \bigcup B_t$, $t \in T$ — объединения элементов $\mathbb{F}$, и обозначить новое отношение R_2. Точнее, для каждой такой пары доопределить функцию принадлежности μ_{R_1} отношения R_1 следующим образом:

$$\mu_{R_2}(X, Y) = \begin{cases} \max\left(\mu_{R_1}(A, B), \min\limits_t\left(\mu_{R_1}(A_t, B_t)\right)\right) \\ \text{при } X = A, Y = B, \\ \mu_{R_1}(X, Y) \text{ иначе}. \end{cases}$$

Полученное отношение R_2 объединить с отношением $\supseteq$.

Теорема 1.1. Логическое замыкание нечеткого отношения R совпадает с транзитивным замыканием $\tilde{R}^*$ соответствующего отношения $\tilde{R}$.

2. Нечеткие продукционно-логические уравнения

В этом разделе определяется класс уравнений в нечеткой LP-структуре. Излагается основной подход к вычислению решений. Процесс нахождения решения соответствует обратному нечеткому логическому выводу в продукционной системе.

Пусть дано нечеткое отношение R на атомно-порожденной решетке $\mathbb{F}$. Пусть также $\mu_R(A, B) > 0$ для некоторых элементов $A, B \in \mathbb{F}$. Тогда B является образом A, а A — прообразом B при отношении R. При этом каждый элемент решетки может иметь множество образов и прообразов с различной степенью принадлежности (значением $\mu_R(A, B)$).

Для данного $B \in \mathbb{F}$ *минимальным прообразом* при отношении R называется такой элемент $A \in \mathbb{F}$, что $\mu_R(A, B) > 0$ и A минимален, т. е. не содержит другого $A_1 \in \mathbb{F}$, для которого $\mu_R(A_1, B) > 0$.

Определение 2.1. Атом $x \in \mathbb{F}$ называется начальным при нечетком отношении R, если нет ни одной пары $A, B \in \mathbb{F}$ такой, что $\mu_R(A, B) > 0$, причем x содержится в B и не содержится в A. Элемент X называется начальным, если все его атомы начальны. Подмножество $\mathbb{F}_0(R)$ (иногда для краткости обозначаем $\mathbb{F}_0$), состоящее из всех начальных элементов $\mathbb{F}$, называется начальным множеством решетки $\mathbb{F}$ при отношении R.

Пусть $\bar{R}$ — логическое замыкание отношения R. Согласно его структуре [12], множества $\mathbb{F}_0(R)$ и $\mathbb{F}_0(\bar{R})$ совпадают.

Рассматривается уравнение

$$\bar{R}(X) = B, \tag{1}$$

где $B \in \mathbb{F}$ — заданный элемент; $X \in \mathbb{F}$ — неизвестный.

Определение 2.2. *Приближенным решением* уравнения (1) называется любой прообраз элемента B в $\mathbb{F}_0$ (при отношении $\bar{R}$). *Решением (точным)* уравнения (1) называется любой минимальный прообраз элемента B в $\mathbb{F}_0$. *Общим решением* уравнения (1) называется совокупность всех его решений $\{X_s\}$, $s \in S$.

Уравнения вида (1) называются *продукционно-логическими уравнениями* в нечеткой LP-структуре.

Замечание 2.1. По определению, точное решение уравнения (1) является и приближенным. Приближенное решение всегда содержит хотя бы одно точное решение.

Обстоятельство, затрудняющее процесс решения уравнения (1), состоит в том, что исходно известно лишь отношение R. Оно в моделируемой интеллектуальной системе соответствует заданному множеству продукций — нечеткой базе знаний. Однако решение уравнения требуется найти как прообраз его правой части при отношении $\bar{R}$ — логическом замыкании R. При этом построение логического замыкания нецелесообразно, поскольку на практике требует значительный объем ресурсов (вычислительных и занимаемой памяти).

Нечеткость отношения R вносит дополнительный фактор, усложняющий методы решения уравнения (1) и саму постановку этой задачи. Помимо требуемой минимальности искомого прообраза X, необходимо учитывать и значение функции принадлежности $\mu_{\bar{R}}(X, B)$.

В настоящей работе рассматривается наиболее простая постановка задачи для уравнения (1). Требуется найти все решения, на которых функция принадлежности принимает положительные значения. Для каждого полученного решения X значение $\left(\mu_{\bar{R}}(X, B)\right)$ должно быть вычислено.

Далее решается вопрос о том, каким образом меняется общее решение уравнений вида (1) при объединении их правых частей. Точнее наоборот, можно ли для нечеткого отношения R вместо исходного уравнения решить несколько уравнений с более простыми правыми частями. Ответ дает следующая теорема. Ее доказательство планируется к опубликованию в отдельной работе.

Рассмотрим уравнение

$$\bar{R}(X) = B_1 \cup B_2. \tag{2}$$

Теорема 2.1. Пусть $\{X_p\}$, $p \in P$ — общее решение уравнения вида (1) с правой частью B_1, а $\{Y_q\}$, $q \in Q$ — общее решение уравнения того же вида с правой частью B_2. Тогда общее решение уравнения (2) представляет собой множество всех элементов вида $X_p \cup Y_q$, из которого исключены элементы, содержащие другие элементы этого же множества.

3. Методы эквивалентного упрощения уравнений

Наряду с теоремой 2.1 рассмотрим дальнейшие методы упрощения нечетких уравнений вида (1). В этом разделе приводятся в основном лишь формулировки математических результатов. Их строгие доказательства планируются к опубликованию в отдельной работе.

Будем предполагать, что R является конечным каноническим нечетким отношением на решетке $\mathbb{F}$, не содержащим пар отношения $\supseteq$, а правая часть B уравнения (1) представляет конечное объединение атомов. Согласно работам [12, 13] отношение, не удовлетворяющее указанным условиям, может быть эквивалентно преобразовано в таковое.

Определим расслоение нечеткого отношения R на виртуальные слои $\{R^t \mid t \in T\}$. В отдельных слоях упрощается построение и исследование алгоритмов, связанных с решением уравнения.

Построение слоев целесообразно начать в терминах четких множеств. Рассмотрим носитель R — четкое отношение S_R. Напомним, что это четкое множество всех упорядоченных пар элементов решетки, на которых функция принадлежности μ_R принимает положительные значения. Разобьем его на непересекающиеся подмножества, каждое из которых образовано всеми парами $\left(A, x_p\right) \in S_R$ с одним и тем же атомом x_p в качестве правой части. Такое разбиение возможно, поскольку исходное отношение R является каноническим. Обозначим эти подмножества $S_R(p)$ по их атому x_p, $p \in P$.

В соответствии с работой [6] слоем S_R^t в отношении S_R называется его подмножество, образованное упорядоченными парами, взятыми по одной из каждого непустого $S_R(p)$, $p \in P$. Для отношения S_R справедливы свойства, следующие из способа построения множества $\{S_R^t \mid t \in T\}$.

Замечание 3.1. Каждый слой S_R^t содержит максимально возможное подмножество пар в S_R с уникальными правыми частями. При добавлении к слою еще одной пары из S_R свойство уникальности нарушится.

Замечание 3.2. Любое подмножество пар в S_R с уникальными правыми частями содержится в некотором слое.

Замечание 3.3. В общем случае слои имеют непустые пересечения. Объединение всех слоев равно S_R.

Замечание 3.4. Общее число слоев $|T|$ определяется равенством $|T| = \prod_{p \in P} |T|_p$, где $|T|_p$ — мощность подмножества пар отношения S_R с правой частью x_p.

Возвращаясь к нечеткому отношению R, сформулируем следующее определение.

Определение 3.1. Слоем R^t в нечетком отношении R называется отношение, определяемое следующей функцией принадлежности:

$$\mu_{R^t}(A,b) = \begin{cases} \mu_R(A, b), \text{ при } (A,b) \in S_R^t, \\ 0 \text{ иначе}. \end{cases}$$

Согласно сделанным построениям $R = \bigcup_t R^t$.

Замечания 3.1—3.4 имеют аналоги для случая нечеткого отношения R и его расслоения $\{R^t \mid t \in T\}$.

Пусть $A \in \mathbb{F}$. Будем говорить, что элемент $B \in \mathbb{F}$ получен из A применением пары (Y, z), если $\mu_R(Y, z) > 0$, $Y \subseteq A$ и $B = A \cup z$. При этом очевидно, что $\mu_{\bar{R}}(A, B) \geqslant \mu_R(Y, z)$.

Рассмотрим некоторые свойства построенного выше расслоения $\{R^t \mid t \in T\}$ для отношения R. В частности, окажется полезным следующее утверждение. Его доказательство приводится, поскольку будет использовано в дальнейшем изложении.

Лемма 3.1. Пусть $\mu_{\bar{R}}(A, B) > 0$. Тогда для вывода этого значения существует кортеж $(C_1, ..., C_m)$ элементов решетки $\mathbb{F}$ такой, что $A = C_0 \subset C_1 \subset ... \subset C_m \supseteq B$, причем каждый элемент $C_j, j > 0$ получен из C_{j-1} применением некоторой пары (X_j, z_j). При $\mu_R(A, B) > 0$ полагаем $m = 0$.

Доказательство. В условиях леммы по теореме 1.1 имеется упорядоченный конечный набор элементов $(B_1, ..., B_n)$ такой, что в последовательности (B_0, B_1), (B_1, B_2), ..., (B_n, B_{n+1}), где $B_0 = A$, $B_{n+1} = B$, для каждой пары (B_j, B_{j+1}) справедливо $\mu_{\tilde{R}}(B_j, B_{j+1}) > 0$ (см. разд. 1), причем $\mu_{\bar{R}}(A, B) \geqslant \min_{0 \leqslant j \leqslant n} \mu_{\tilde{R}}(B_j, B_{j+1})$. При $\mu_{\tilde{R}}(A, B) > 0$ считаем, что $n = 0$.

Заметим, что по построению отношения $\tilde{R}$ имеет место $\mu_{\tilde{R}}(B_j, B_j \cup B_{j+1}) \geqslant \mu_{\tilde{R}}(B_j, B_{j+1}) > 0$ для всех $j = 0, ..., n$. Применяя это свойство последовательно по $j = 0, ..., n$, получим новый набор $(\tilde{B}_1, ..., \tilde{B}_n)$, где $\tilde{B}_k = \bigcup_{j=0,...,k} B_j$, $k = 0, ..., n$. Данный кортеж, как и исходный $(B_1, ..., B_n)$, также реализует транзитивную связь элементов решетки A и B. При этом он не уменьшает выводимое значение $\mu_{\bar{R}}(A, B)$ и обладает дополнительным свойством монотонности $A = \tilde{B}_0 \subset \tilde{B}_1 \subset ... \subset \tilde{B}_n \supseteq B_{n+1} = B$.

Рассмотрим теперь структуру пар $(\tilde{B}_{j-1}, \tilde{B}_j)$, $j = 1, ..., n$. Учитывая содержание процесса построения отношения $\tilde{R}$ (разд. 1), нетрудно заметить следующий факт. Отношение R_1 (это R, дополненное отношением рефлексивности на решетке), образует базис при построении $\tilde{R}$ на основе операции объединения. Другими словами, каждое значение $\mu_{\tilde{R}}(\tilde{B}_{j-1}, \tilde{B}_j) > 0$ может быть выведено из некоторого (рефлексивно-дополненного) нечеткого подмножества R_j отношения R путем объединения левых (для $\tilde{B}_{j-1}$) и правых (для $\tilde{B}_j$) частей упорядоченных пар. Отсюда следует, что элемент $\tilde{B}_j$, включающий $\tilde{B}_{j-1}$, может быть получен из $\tilde{B}_{j-1}$ последовательным применением пар отношения $R_j \subseteq R$. Распространяя данное свойство на каждую пару $(\tilde{B}_{j-1}, \tilde{B}_j)$, $j = 1, ..., n$, при этом исключая избыточные (приносящие меньшие выводимые значения $\mu_{\bar{R}}$) применения (рефлексивно-дополненных) пар R, приходим к утверждению леммы при некотором $m \geqslant n$.

Замечание 3.5. Построенный выше кортеж $(C_0, ..., C_m)$ можно выбрать *точным* в следующем смысле. Правый атом z_j каждой примененной в кортеже для перехода от C_{j-1} к C_j пары (Y_j, z_j) должен содержаться в B либо в левой части Y_{j+k} некоторой применяемой позднее пары (Y_{j+k}, z_{j+k}), и не может содержаться в C_{j-1}. Переходы, не удовлетворяющие такому свойству, не являются необходимыми для получения B. Поэтому они могут быть исключены из кортежа вместе с C_j.

Далее приведем свойства, характеризующие расслоение $\{R^t\}$ отношения R с точки зрения решения уравнений вида (1).

Лемма 3.2. Если $\mu_{\bar{R}}(A, B) > 0$, то в R существует слой R^t такой, что $\mu_{\bar{R}^t}(A, B) = \mu_{\bar{R}}(A, B)$. Здесь $\bar{R}^t$ — логическое замыкание отношения R^t.

Следствие 3.1. Логическое замыкание канонического нечеткого отношения R равно объединению логических замыканий его слоев, т. е. $\bar{R} = \bigcup_{t \in T} \bar{R}^t$.

Будем говорить, что решение X уравнения (1) (точное или приближенное) порождается в R некоторым слоем R^t, если $\mu_{\bar{R}}(X, B) = \mu_{\bar{R}^t}(X, B)$.

Замечание 3.6. Согласно лемме 3.2 любое решение уравнения (1) порождается в R некоторым слоем R^t.

Замечание 3.7. Очевидно, что для нахождения решения уравнения (1) в слое R^t достаточно вместо (1) решить аналогичное уравнение с отношением $\bar{R}^t$.

Однако два различных слоя в общем случае могут порождать одно и то же решение. Кроме того, могут существовать слои, дающие точное решение в R^t,

но приближенное в R. Некоторые слои могут вообще не давать решений. Однако справедливо утверждение о том, что один слой не может порождать более одного точного решения.

Лемма 3.3. Ни один слой R^t отношения R не может порождать двух различных точных решений уравнения (1).

Объединяя приведенные выше результаты, можно сформулировать следующую теорему.

Теорема 3.1. Для нахождения общего решения уравнения (1) достаточно найти (единственное) решение X_t в каждом слое R^t, в котором оно существует. Далее из полученного множества решений необходимо исключить элементы, содержащие другие элементы этого же множества.

Следствие 3.2. Число точных решений уравнения (1) оценивается сверху выражением $M = \prod_{p \in P} M_p$

(см. замечание 3.4).

Теорема 3.1 и замечание 3.7 позволяют свести вопрос о решении уравнения (1) к задаче нахождения решения уравнений вида

$$\bar{R}^t(X) = B, \qquad (3)$$

где B — неначальный элемент решетки $\mathbb{F}$; R^t — произвольный слой в R.

В следующем разделе рассмотрен вопрос о существовании решений нечетких продукционно-логических уравнений на решетке и обоснована заключительная стадия их нахождения.

4. Существование и способ вычисления решений

Основываясь на теоремах 2.1 и 3.1, для решения исходного уравнения (1) достаточно решить совокупность уравнений в слоях с каждым атомом элемента B в качестве правой части. Поэтому рассмотрим задачу нахождения решения следующего уравнения:

$$\bar{R}^t(X) = b, \qquad (4)$$

где b — неначальный атом решетки $\mathbb{F}$; R^t — слой в R.

Покажем, что она эквивалентна задаче на ориентированном графе перечисления всех входных вершин, из которых достижима заданная вершина. Вначале построим такой граф G_{R^t}.

Пусть (Y, z) — произвольная пара элементов решетки, для которой $\mu_{R^t}(Y, z) > 0$. Атому z и каждому атому элемента Y поставим в соответствие вершину графа G_{R^t}. Далее построим дуги, ведущие из всех вершин, соответствующих атомам элемента Y, в вершину, соответствующую z. Эту вершину пометим значением $\mu_{R^t}(Y, z)$. Аналогичные построения выполним для всех таких пар. Начальные вершины графа пометим значением 1. Для краткости изложения иногда будем отождествлять атомы решетки и соответствующие им вершины графа.

В полученном графе G_{R^t} выберем вершину b, соответствующую правой части уравнения (4). Рассмотрим подграф $G_{R^t,b} \subseteq G_{R^t}$, состоящий из b и всех вершин G_{R^t}, из которых она достижима.

Лемма 4.1. Если граф $G_{R^t,b}$ не содержит ориентированных циклов, то множество всех его начальных вершин соответствует точному решению уравнения (4).

Доказательство. Пусть выполнено условие леммы. Покажем, как с помощью графа $G_{R^t,b}$ построить решение X уравнения (4) и точный кортеж вывода $(C_0, ..., C_m)$ значения $\mu_{\bar{R}^t}(X, b) > 0$ (см. замечание 3.5). Проведем построение этого кортежа индуктивно справа налево.

Формирование C_m начнем с включения в него единственного атома — $b_m = b$, далее переходим к построению предшествующего элемента C_{m-1}. Опишем действия, выполняемые на очередном шаге $j = m, m - 1, ..., 0$. Если все атомы текущего элемента C_j окажутся начальными, то процесс построения кортежа будем считать завершенным ($j = 0$). В противном случае выберем произвольный неначальный атом $b_j \subseteq C_j$. Обозначим $\{b^k_{j-1} \mid k = 1,..., N_{j-1}\}$ — множество всех атомов, дуги из которых в графе $G_{R^t,b}$ непосредственно входят в вершину b_j. В элементе C_j (как объединении атомов) заменим b_j объединением указанных $\{b^k_{j-1}\}$ и обозначим полученный элемент C_{j-1}. Наконец, скорректируем все пройденные $C_j, ..., C_m$, присоединяя к ним атомы $\{b^k_{j-1}\}$, получая таким образом теоретически нестрого монотонную последовательность $C_{j-1} \subseteq C_j \subseteq ... \subseteq C_m$. Однако в нашем случае в этой цепочке равенства невозможны. Повторное рассмотрение атома b_j и возвращение его в C_{j-1} на последующих шагах означало бы наличие цикла в графе $G_{R^t,b}$.

По условию леммы граф $G_{R^t,b}$ конечен и не содержит ориентированных циклов, поэтому описанный выше процесс переходов "против стрелок" также конечен. Результирующий кортеж $(C_0, ..., C_m)$ является точным в смысле замечания 3.5. Более того, элемент $X = C_0$ представляет собой точное решение уравнения (4). Дело в том, что по построению X все его атомы участвуют в выводе значения $\mu_{\bar{R}^t}(X, b) > 0$. Этот вывод детерминирован с точностью до порядка взаимно-независимых переходов кортежа вывода.

Описанным способом, очевидно, будут пройдены все вершины графа $G_{R^t,b}$. Таким образом, можно утверждать, что X состоит из *всех* начальных атомов решетки, соответствующих начальным вершинам графа $G_{R^t,b}$. В силу принятой в настоящей работе (max-min)-композиции для нечетких отношений, значение $\mu_{\bar{R}^t}(X, b)$ вычисляется как наименьшее значение из меток всех вершин $G_{R^t,b}$.

Лемма 4.2. Если в графе $G_{R^t,b}$ есть ориентированный цикл, то уравнение (4) не имеет решений.

Доказательство. Предположим противное, что при наличии цикла существует решение X уравнения (4), и построим соответствующий ему точный кортеж $(C_0, ..., C_m)$. Пусть y_1 — некоторая вершина графа $G_{R^t,b}$, содержащаяся в его ориентированном цикле. Такая вершина не может соответствовать начальному атому по его определению, поэтому должна существовать пара (Y_1, y_1), по построению графа $G_{R^t,b}$ участвующая в выводе значения $\mu_{\bar{R}^t}(X, b)$. Таким

образом, один из переходов (C_k, C_{k+1}) в кортеже осуществляется парой (Y_1, y_1).

Далее рассмотрим вершину y_2, предшествующую y_1 в цикле графа. Соответственно имеем $y_2 \in Y_1$, поскольку по построению графа элемент Y_1 содержит все вершины, непосредственно предшествующие y_1. Для вершины y_2 справедливы все предыдущие рассуждения, связанные с y_1. Соответственно один из предшествующих элементам (C_k, C_{k+1}) переходов в кортеже должен осуществляться некоторой парой (Y_2, y_2).

Продолжая такое рассмотрение, за каждый шаг будем перемещаться в кортеже как минимум на одну позицию влево, а в графе — не более чем на одну вершину назад в цикле. Через конечное число шагов выйдем за начало кортежа, т. е. очередная рассматриваемая вершина y_i будет соответствовать атому в X, но в графе вершина y_i останется в том же цикле. Данное обстоятельство означает противоречие, так как в X содержатся только начальные атомы, а в цикле графа — лишь невходные вершины.

Объединяя утверждения доказанных лемм 4.1 и 4.2, можно сформулировать следующий факт, завершающий обоснование пошагового процесса решения уравнения (4).

Теорема 4.1. Уравнение (4) имеет не более одного решения. Если граф $G_{R^t,b}$ не содержит ориентированных циклов, то единственное решение состоит из всех атомов, соответствующих начальным вершинам графа, при этом значение функции принадлежности $\mu_{\overline{R}^t}(X, b)$ вычисляется как наименьшее из меток вершин. Если $G_{R^t,b}$ содержит ориентированный цикл, решений нет.

5. О нечетком LP-выводе и показателях релевантности

Изложим некоторые идеи практического применения аппарата нечетких продукционно-логических уравнений для оптимизации нечеткого обратного вывода (релевантного FLP-вывода).

Как известно [7], стратегия релевантного LP-вывода направлена на минимизацию числа медленно выполняемых запросов (к базе данных или интерактивному пользователю). Ее основная цель, чтобы запросы по возможности соответствовали тем фактам, которые действительно необходимы при выводе. Отрицательный ответ на единичный запрос исключает последующие запросы об элементах целого подмножества фактов, соответствующего прообразу проверяемой гипотезы (объекта экспертизы). Кроме того, при LP-выводе предпочтение отдается тестированию множеств фактов минимальной мощности.

Первая стадия обратного вывода на основе продукционно-логического уравнения состоит в его решении — построении всех минимальных начальных прообразов в LP-структуре для атомов, соответствующих значениям объекта экспертизы. Далее в построенном множестве достаточно найти тот прообраз, который содержит лишь истинные факты, после чего сразу можно сделать заключение о соответствующем значении объекта экспертизы.

Простой путь в этом направлении — просматривать прообразы последовательно, задавая пользователю вопросы о соответствующих начальных фактах (или обращаясь за ними к базе данных). Этот способ уже дает преимущества — исследуются лишь минимальные прообразы. Также предварительно можно исключить из процесса "противоречивые" прообразы, т. е. содержащие одновременно альтернативные значения одного и того же объекта.

Однако существует более эффективный способ — приоритетный просмотр прообразов, содержащих значения наиболее "релевантных" объектов. Таковыми в первую очередь считаются объекты, чьи значения присутствуют в максимальном числе построенных прообразов. Тогда единственный отрицательный ответ на заданный вопрос исключает из рассмотрения сразу большое число прообразов, что ускоряет исследование. Второй показатель релевантности тестируемого объекта — наличие его значений в прообразах минимальной мощности. Таким образом, предпочтение отдается тем прообразам, проверка истинности которых потребует меньшего числа вопросов пользователю (или обращений к базе данных).

Сочетанием указанных двух показателей релевантности можно достичь результатов, по эффективности существенно превышающих результаты, возможные в случае классической машины вывода. Как показывают эксперименты [7], с помощью релевантного LP-вывода можно достичь снижения числа выполняемых медленных запросов в среднем на 15...20 %.

Нечеткий характер интеллектуальной системы создает дополнительные задачи и возможности для повышения эффективности обратного логического вывода. Одной из целей моделирования нечеткой продукционной системы является достижение за приемлемое время для выводимой гипотезы более высокого значения коэффициента уверенности. Таким образом, при реализации стратегий нечеткого релевантного LP-вывода (FLP-вывода), наряду с отмеченными выше двумя характеристиками, необходимо учитывать дополнительный показатель релевантности. Он представляет значение функции принадлежности, вычисляемое для каждого найденного решения уравнения вида (1).

Здесь возможны теоретические исследования, например, на основе методов многокритериальной оптимизации. Вместе с тем имеется широкое поле для практических экспериментов, комбинирующих три показателя релевантности с различными "весами", выбираемыми соответственно поставленным задачам в конкретной предметной области.

Заключение

В настоящей работе рассмотрен класс продукционно-логических уравнений в нечеткой LP-структуре, расширяющей область применения данного подхода до моделирования нечетких интеллектуальных систем продукционного типа.

Сформулированы методы решения этих уравнений. Впервые решены вопросы о существовании и числе решений. Нахождение решения продукционно-логического уравнения соответствует обратному

нечеткому логическому выводу в продукционной системе. Доказанные теоремы могут быть использованы для программной реализации нечетких LP-структур и соответствующей оптимизации нечеткого логического вывода.

Одним из следующих шагов на рассматриваемом направлении является практическое исследование стратегий FLP-вывода, использующих новый дополнительный показатель релевантности, характеризующий коэффициенты уверенности проверяемых при выводе гипотез.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00037.

Список литературы

1. **Oles F. J.** An Application of Lattice Theory to Knowledge Representation // Theor. Comput. Sci. Oct. — 2000. — Vol. 249, No. 1. — P. 163—196.

2. **Бениаминов Е. М.** Алгебраические методы в теории баз данных и представлении знаний. — М.: Научный мир, 2003. — 184 с.

3. **Жожикашвили А. В., Стефанюк В. Л.** Алгебраическая теория продукционных систем // VIII национальная конференция по искусственному интеллекту с международным участием КИИ-2002. 7—12 октября 2002, г. Коломна: Труды конференции. Т. 1. — М.: Физматлит, 2002. — С. 428—436.

4. **Maciol A.** An application of rule-based tool in attributive logic for business rules modeling Expert Systems with Applications. — 2008. — Vol. 34, No. 3. — P. 1825—1836.

5. **Дородных Н. О., Юрин А. Ю.** Использование диаграмм классов UML для формирования продукционных баз знаний // Программная инженерия. — 2015. № 4. — С. 3—9.

6. **Махортов С. Д.** Математические основы искусственного интеллекта: теория LP-структур для построения и исследования моделей знаний продукционного типа / Под ред. В. А. Васенина. — М.: Изд-во МЦНМО, 2009. — 304 с.

7. **Болотова С. Ю., Махортов С. Д.** Алгоритмы релевантного обратного вывода, основанные на решении продукционно-логических уравнений // Искусственный интеллект и принятие решений. — 2011. — № 2. — С. 40—50.

8. **Махортов С. Д.** Продукционно-логические уравнения в распределенной LP-структуре // Программная инженерия. — 2016. — Т. 7, № 7. — С. 324—329.

9. **Батыршин И. З., Недосекин А. О., Стецко А. А.** и др. Нечеткие гибридные системы: Теория и практика / Под ред. Н. Г. Ярушкиной. — М.: Физматлит, 2007. — 208 с.

10. **Махортов С. Д., Шмарин А. Н.** Нечеткий LP-вывод и его программная реализация // Программная инженерия. — 2013. — № 12. — С. 34—38.

11. **Махортов С. Д., Шмарин А. Н.** Оптимизация метода LP-вывода // Нейрокомпьютеры. Разработка, применение. — 2013. — № 9. — С. 59—63.

12. **Махортов С. Д.** Алгебраическая модель интеллектуальной системы с нечеткими правилами // Программная инженерия. — 2019. — Т. 10, № 11—12. — С. 457—463.

13. **Махортов С. Д., Клейменов И. В.** О логической редукции алгебраической модели интеллектуальной системы с нечеткими правилами // Вестник Воронежского государственного университета. Серия: Физика. Математика. — 2019. — № 3. — С. 67—78.

14. **Биркгоф Г.** Теория решеток: пер. с англ. — М.: Наука, 1984. — 568 с.

15. **Рыжов А. П.** Элементы теории нечетких множеств и ее приложений. — М.: Диалог-МГУ, 2003. — 81 с.

16. **Garmendia L., Del Campo R. G., López V., Recasens J.** An Algorithm to Compute the Transitive Closure, a Transitive Approximation and a Transitive Opening of a Fuzzy Proximity // Mathware & Soft Computing. — 2009. — Vol. 16. — P. 175—191.

On the Solvability and Number of Solutions of Production-Logical Equations in a Fuzzy LP-Structure

S. D. Makhortov, msd_exp@outlook.com, Voronezh State University, Voronezh, 394018, Russian Federation

Corresponding author:

Makhortov Sergey D., Head of Department, Voronezh State University, Voronezh, 394018, Russian Federation
E-mail: msd_exp@outlook.com

Received on August 12, 2020
Accepted on September 01, 2020

For the construction and study of formal models of intelligent information systems, algebraic methods are useful. One of the topical directions here is the production-type logical systems, which are widespread in computer science.

In recent years, the author and his followers have been developing the algebraic theory of LP-structures (lattice production structures). It is designed to formalize and solve a number of knowledge management problems in production systems. The method of relevant backward inference (LP-inference) was also introduced and investigated, which significantly reduces the number of calls to external information sources in comparison with classical inference. Subsequently, the theory was generalized to speed up inference in distributed production systems.

At the same time, modern intelligent systems are characterized by fuzzy knowledge and fuzzy reasoning. Therefore, there is a need to extend the theory of LP-structures to fuzzy production systems. This research was initiated in previous articles by the author. Some concepts are introduced that impart fuzziness to LP-structures, and certain properties of fuzzy LP-inference are established. In recent works, research results are presented that systematically generalize the theory of LP-structures to the case of fuzzy knowledge bases. The terminology of FLP-structures with fuzzy logical relation (Fuzzy LP-structures) is introduced, the main standard properties are proved.

The present work complements the FLP-structure model by investigating a class of production-logical equations. Relevant inference ideas are based on it, reducing the number of calls to external sources of information.

Methods for solving these equations are formulated. For the first time, questions about the existence and number of solutions have been resolved. Finding a solution to a production-logical equation corresponds to the backward fuzzy inference in a production system. The proved theorems can be used for software implementation of fuzzy LP-structures and corresponding optimization of fuzzy inference. Some ideas for this implementation are discussed.

Keywords: *fuzzy production system, relevant backward inference, algebraic model, fuzzy LP-structure, production-logical equations*

For citation:

Makhortov S. D. On the Solvability and Number of Solutions of Production-Logical Equations in a Fuzzy LP-Structure, *Programmnaya Ingeneria,* 2021, vol. 12, no. 1, pp. 40—47

DOI: 10.17587/prin.12.40-47

References

1. **Oles F. J.** An Application of Lattice Theory to Knowledge Representation, *Theor. Comput. Sci.*, Oct. 2000, vol. 249, no. 1, pp. 163—196.

2. **Beniaminov E. M.** *Algebraic methods in the theory of databases and knowledge representation*, Nauch nyj mir, 2003, 184 p. (in Russian).

3. **Zhozhikashvili A. V., Stefanjuk V. L.** Algebraic theory of production systems, *VIII nacional'naja konferencija po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2002*, Kolomna, October 7—12, 2002, Trudy konferencii. T. 1. Fizmatlit, 2002, pp. 428—436 (in Russian).

4. **Maciol A.** An application of rule-based tool in attributive logic for business rules modeling, *Expert Systems with Applications,* 2008, vol. 34, no. 3, pp. 1825—1836.

5. **Dorodnykh N. O., Yurin A. Yu.** The Use of Diagrams of UML Classes for Production Knowledge Bases Formation, *Programmnaya Ingeneria*, 2015, no. 4, pp. 3—9 (in Russian).

6. **Makhortov S. D.** *Mathematical Foundations of Artificial Intelligence: The LP structures theory for the knowledge models of production type construction and research* / Eds. by V. A. Vasenin. Moscow, MCCME, 2009, 304 p. (in Russian).

7. **Bolotova S. Yu., Makhortov S. D.** Algorithms of the relevant backward inference that is based on production-logic equations solving, *Iskusstvennyiy intellekt i prinyatie resheniy*, 2011, no. 2, pp. 40—50 (in Russian).

8. **Makhortov S. D.** Production-logical equations in the distributed LP-structure, *Programmnaya Ingeneria*, 2016, vol. 7, no. 7, pp. 324—329 (in Russian).

9. **Batyrshin I. Z., Nedosekin A. O., Stecko A. A.** et al. *The Fuzzy Gibrid Systems: Theory and Practice:* Fizmatlit, 2007, 208 p. (in Russian).

10. **Makhortov S. D., Shmarin A. N.** Fuzzy LP-inference and its software implementation, *Programmnaya Ingeneria,* 2013, no. 12, pp. 34—38 (in Russian).

11. **Makhortov S. D., Shmarin A. N.** Optimizing LP-inference method, *Nejrokomp'jutery. Razrabotka, primenenie*, 2013, no. 9, pp. 59—63 (in Russian).

12. **Makhortov S. D.** An Algebraic Model of the Intelligent System with Fuzzy Rules. *Programmnaya Ingeneria*, 2019, vol. 10, no. 11—12, pp. 457—463 (in Russian).

13. **Makhortov S. D., Kleymenov I. V.** On Logical Reduction of an Algebraic Model of the Intellectual System with Fuzzy Rules, *Vestnik VGU. Serija Fizika. Matematika*, Voronezh, 2019, no. 3, pp. 67—78 (in Russian).

14. **Birkhoff G.** *Lattice Theory,* Rhode Island, 1995, 420 p.

15. **Ryzhov A. P.** *Elements of the Theory of Fuzzy Sets and of its Applications,* Dialog-MGU, 2003, 81 p. (in Russian).

16. **Garmendia L., Del Campo R. G., López V., Recasens J.** An Algorithm to Compute the Transitive Closure, a Transitive Approximation and a Transitive Opening of a Fuzzy Proximity, *Mathware & Soft Computing*, 2009, no. 16, pp. 175—191.

ИНФОРМАЦИЯ

УДК 004.75 + 004.41 DOI:10.17587/prin.12.48-57

А. С. Шундеев, канд. физ.-мат. наук, вед. науч. сотр., alex.shundeev@gmail.com,
Д. Д. Заславский, студент, zabaf@ya.ru, **С. И. Пехтерев**, студент, stas-19000@mail.ru,
Московский государственный университет им. М. В. Ломоносова

Уменьшение размерности векторного представления документов с помощью метода главных компонент

Векторные представления слов являются популярным объектом исследований, и эффективным средством анализа текстовых данных начиная с 1970-х годов. В частности, с их помощью удалось формализовать и дать решения задачам определения смысловой близости слов и поиска аналогий. В данной статье векторные представления слов рассматриваются с точки зрения решения задачи классификации текстовых документов.

Популярным средством, используемым для снижения размерности данных, является метод главных компонент. В том числе он применяется и к векторным представлениям слов. В последнее время появился ряд работ, в которых исследуется не совсем традиционный подход к применению данного метода. В них предлагается удалять проекции не на последние, а на первые главные компоненты. Проводимые на этом направлении экспериментальные исследования показывают, что точность решения задач определения смысловой близости слов и поиска аналогий при этом может увеличиться. Целью исследования, описанного в данной работе, является проверка того, сохраняется ли подобный эффект при решении задачи классификации текстовых документов.

Ключевые слова: *векторное представление слов, векторное представление документов, классификация текстов, Word2Vec, GloVe, fastText, дистрибутивная гипотеза, пост-обработка, метод главных компонент*

Введение

Современным и бурно развивающимся подходом в области анализа текстовых данных является использование так называемых векторных представлений слов (*word embeddings*), которые выступают основным объектом исследования в рамках данной работы. Векторное представление слов устанавливает собой соответствие между словами и вещественными векторами фиксированной размерности. При построении такого представления пытаются добиться следующей цели. Близким по смыслу словам должны соответствовать близкие векторы. Подобное построение осуществляется в рамках некоторой модели, которая имеет ряд настраиваемых параметров и основывается на обработке входного корпуса текстов. Процесс построения векторного представления слов именуют также как процесс обучения.

Проиллюстрируем введенные понятия на примере. В открытом доступе имеется целый ряд построенных наборов векторных представлений слов, которые можно использовать в образовательных и исследовательских целях. Как правило, подобные наборы предоставляются авторами соответствующих моделей векторных представлений слов и призваны продемонстрировать их преимущества. В частности, создатели модели GloVe (*Global Vectors for Word Representation*) [1] подготовили наборы[1] векторных представлений слов размерностей 50, 100, 200 и 300, которые были получены в результате обработки текстов статей англоязычной Википедии 2014 г. На рисунке изображены проекции нескольких 50-мерных векторов из этих наборов на двухмерную плоскость. Каждая проекция (точка) подписана соответствующим вектору словом. Невооруженным глазом видно, что близкие по смыслу слова сгруппированы рядом друг с другом.

Следует более формально определить, как при работе с векторными представлениями слов определяется близость между вещественными векторами и что понимается под смысловой близостью слов.

На практике используются разные подходы для определения близости между векторами, в том числе косинусная близость. В работе [2] на примере решения ряда задач показывается, что наилучшие результаты достигаются с использованием косинусной близости (косинус угла между векторами). Значение косинусной близости, равное единице, трактуется как максимальное сходство между векторами и соответствующими им словами, а нулевое значение трактуется как максимальное различие. Так, например, для слова *red* наиболее близкими оказываются

[1] https://nlp.stanford.edu/projects/glove

Разбиение набора слов на смысловые группы

слова *yellow* (0,899), *blue* (0,890), *green* (0,856), *black* (0,840), *purple* (0,832). В скобках указано значение косинусной близости. Для слова *cat* наиболее близкими оказываются слова *dog* (0,921), *rabbit* (0,848), *monkey* (0,804), *rat* (0,789), *cats* (0,786).

В компьютерной лингвистике считается, что два слова являются семантически близкими (*semantically similar*) [3], если они имеют общую родительскую категорию (гипероним, "сверх-имя"). Так, *собака* и *кролик* оба являются *животными*. *Москва*, *Берлин*, *Вена* являются *столицами*. Более общим понятием является семантическая связность (*semantic relatedness*) слов [4]. Семантическая связность включает в себя такие отношения между словами, как синонимия (например, слова *смелый* и *храбрый*), антонимия (например, слова *выигрыш* и *проигрыш*), меронимия (отношение части и целого), гипонимия (родовидовое отношение). К семантически связанным словам также относят слова, которые совместно встречаются в рассматриваемом корпусе текстов.

Как было продемонстрировано выше, векторные представления слов могут успешно использоваться для решения задачи определения смысловой близости между словами. При этом дается числовая оценка смысловой близости двух слов. Кроме того, векторные представления слов могут успешно использоваться для изучения смысловых отношений между словами. Например, пары слов (*Москва*, *Россия*), (*Берлин*, *Германия*) и (*Минск*, *Белоруссия*) являются примерами отношения "столица — страна". Подобные пары слов называют аналогиями [5]. В работах [6, 7] была поставлена задача поиска аналогий, имеющая следующую формулировку. Для заданных слов *A*, *B*, *C* необходимо подобрать слово *D* таким образом, чтобы пары слов (*A*, *D*) и (*C*, *D*) являлись аналогиями по отношению друг к другу. Например, для слов *Москва*, *Россия*, *Берлин* оптимальным решением будет слово *Германия*. В ходе поиска подходов для решения этой задачи были разработаны две модели векторных представлений слов под общим названием Word2Vec [6, 8].

В настоящее время точность решения задачи определения смысловой близости слов и задачи поиска аналогий является основным критерием качества векторного представления слов.

В области анализа данных существует ряд методов понижения размерности данных [9], представленных в виде вещественных векторов фиксированной размерности. Одним из них является метод главных компонент PCA (*Principal Component Analysis*). Для большинства наборов данных можно построить прямую (ось, направление), обладающую следующим свойством. Проекции векторов из рассматриваемого набора данных на эту прямую будут порождать максимальное рассеивание. Полученную прямую называют первой главной компонентой. Далее описанную процедуру можно применить к подпространству, ортогональному первой главной компоненте. В результате будет получена вторая главная компонента и т. д. Обычно считается, что проекции на последние главные компоненты можно безболезненно удалить из данных. При этом информационное наполнение в этих данных не ухудшится, однако серьезно понизятся накладные расходы по их хранению. В ряде работ, в том числе в работе [10], развивается подход, в рамках которого применительно к векторным представлениям слов необходимо удалять проекции не на последние, а на первые главные компоненты.

В настоящей работе векторные представления слов рассматриваются как основа для построения математических моделей документов. Качество подобных моделей оценивается с точки зрения реше-

ния задачи классификации документов. В качестве базовой модели документа рассматривается среднее арифметическое векторов слов, входящих в его состав [11]. Исследуется вопрос о том, как влияет снижение размерности векторного представления слов, выполненное с применением метода главных компонент, на качество модели документов. В частности, применимы ли выводы, сделанные в работе [10] к векторным представлениям документов.

Настоящая статья является продолжением работ авторов [12—14], в которых исследуются вопросы изменения размерности векторных представлений слов и документов, а также влияние подобных изменений на точность решения задачи классификации документов. В работах [12, 13] был предложен подход, в рамках которого ставилась и решалась задача восстановления многомерной регрессии, позволяющей трансформировать модель и изменить размерность векторного представления корпуса документов. В работе [14] исследовался ряд элементарных моделей векторных представлений документов с точки зрения решения задачи их классификации.

1. Модели векторных представлений слов

Модели векторных представлений слов активно изучаются уже несколько десятилетий. В качестве одной из первых работ на этом направлении можно отметить работу 1975 г. [15]. На начальном этапе подобные модели базировались на построении и преобразовании частотных матриц типа "слово—документ" или "слово—контекст". К этому периоду, в частности, относится создание метода латентного семантического анализа LSA (*Latent Semantic Analysis*) [16], в рамках которого впервые было обосновано применение сингулярного разложения (*Singular Value Decomposition*) частотной матрицы для получения векторного представления слов. Частотные (*count based*) модели подробно описаны в обзорной работе [17].

В последнее время наибольшее внимание уделяется так называемым предсказательным (*predictive*) моделям, в рамках которых векторное представление слов получается как результат решения некоторой оптимизационной задачи. Интерес к предсказательным моделям можно связать с общим ростом популярности нейросетевых методов и подходов. Так, в 2003 г. была разработана вероятностная нейросетевая модель NPLM (*Neural Probabilistic Language Model*) [18]. На протяжении следующих десяти лет эта модель постепенно упрощалась. Результатом таких упрощений стало появление семейства моделей Word2Vec [6, 8], которые представляют собой нейронные сети с одним скрытым слоем, не содержащие нелинейных преобразований.

Преимуществом предсказательных моделей является возможность обучаться на большом объеме исходных текстовых данных. Ответить однозначно, в рамках каких (частотных или предсказательных) моделей можно получить более качественные векторные представления слов, затруднительно [19, 20].

Далее будут рассмотрены две современные предсказательные модели, которые использовались в настоящей работе. Однако прежде чем перейти к их рассмотрению, сформулируем гипотезы из области компьютерной лингвистики, которые положены в основу их построения. Первая гипотеза, получившая название дистрибутивной (*distributional hypothesis*) [21—23], утверждает, что слова, появляющиеся в похожих контекстах внутри корпуса текстов, скорее всего, будут иметь похожий смысл.

В качестве контекста может выступать весь документ или его отдельные фрагменты (предложение, абзац, глава в книге). Часто контекст задается окном некоторого фиксированного размера. Окном является последовательность слов в документе, отстоящих не далее, чем на h позиций от заданного слова w. Само слово w в окно не входит. Контекстом может быть любое слово в окне, множество или мультимножество всех слов окна.

Частотные модели также базируются на гипотезе "мешка слов" (*bag of words hypothesis*) [15], согласно которой смысл документа не зависит от порядка встречающихся в нем слов.

1.1. Модель Word2Vec

Под общим названием Word2Vec скрываются две модели векторных представлений слов [6, 8], а именно — CBOW (*continuous bag-of-words*) и SG (*skip-gram*). В обеих моделях контекст слова задается окном некоторого фиксированного размера h. В качестве примера рассмотрим модель CBOW. Опишем эвристику, которая положена в ее основу.

Рассмотрим следующее предложение:

the quick brown fox jumps over the lazy dog.

Предположим, что $h = 3$. Тогда окно контекста для слова *fox* будет состоять из последовательности слов *the*, *quick*, *brown*, *jumps*, *over*, *the*, как это показано ниже:

the quick brown *fox* ***jumps over the*** *lazy dog.*

Модель CBOW пытается предсказать, какое слово может появиться в заданном контексте. Например, в следующем предложении:

the quick brown *?* ***jumps over the*** *lazy dog,*

на пропущенном месте с большой долей вероятности должно быть восстановлено слово *fox*.

Модели Word2Vec имеют два настраиваемых параметра: размер окна контекста h и размерность целевого векторного представления слов n. В качестве входных данных выступает корпус документов, использующих некоторый словарь D. В ходе построения векторного представления слов для каждого слова $w \in D$ вычисляются два вектора $\mathbf{v}_w, \mathbf{v}'_w \in \mathbb{R}^n$. Итоговое векторное представление слов образуют векторы $\mathbf{v}_w$.

Пусть $h = 1$. Условная вероятность p появления слова w в контексте слова c моделируется с помощью выражения вида

$$p\left(w|c\right) = \frac{\exp\left\langle \mathbf{v}'_w, \mathbf{v}_c \right\rangle}{\sum\limits_{s \in D} \exp\left\langle \mathbf{v}'_s, \mathbf{v}_c \right\rangle}.$$

На основе подобных условных вероятностей записывается функция правдоподобия, для которой решается задача максимизации. Итоговое векторное представление слов является частью решения этой оптимизационной задачи.

Дальнейшее развитие моделей Word2Vec пошло по двум направлениям. В рамках первого направления были разработаны модели векторных представлений документов Doc2Vec [25]. В рамках второго направления была обобщена модель SG, на основе которой была разработана модель векторного представления слов fastText [26], ориентированная на работу с морфологически сложными языками.

1.2. Модель GloVe

Настраиваемыми параметрами модели GloVe [1] являются размерность векторного представления слов n и размер контекста h. По входному корпусу документов строится частотная матрица типа "слово—слово" $\mathbf{X} = (x_{wc})$. О ней также говорят, как о матрице счетчиков совместной встречаемости слов.

В качестве примера в табл. 1 представлена матрица типа "слово—слово", составленная для предложения "*the quick brown fox jumps over the lazy dog*". Контекст задается окном размера $h = 3$.

На основе частотной матрицы $\mathbf{X}$ можно вычислить условную вероятность появления слова $w \in D$ в контексте слова $c \in D$ вида

$$p(w|c) = \frac{x_{wc}}{\sum\limits_{S \in D} x_{ws}}.$$

Авторы [1] делают следующее эвристическое предположение. Пусть заданы три слова $w, c_1, c_2 \in D$. Исследуя отношение условных вероятностей

$$\frac{p(w|c_1)}{p(w|c_2)}$$

можно сделать определенные выводы о семантической связанности этих трех слов друг с другом. Таблица 2 иллюстрирует эти закономерности. В ней в качестве c_1 выбрано слово *ice*, а в качестве c_2 — слово *steam*. В качестве w последовательно выступают слова *solid*, *gas*, *water*, *fashion*. Так, например, если слово w (*solid*) семантически связано с контекстом c_1 и не связано с контекстом c_2, то отношение условных вероятностей будет сравнительно большим. Если наоборот, слово w (*gas*) семантически связано с контекстом c_2 и не связано с контекстом c_1, то отношение будет мало. Если одновременно слово w (*gas*, *fashion*) семантически связано или не связано с контекстами c_1 и c_2, то отношение условных вероятностей будет приблизительно равно единице.

Таблица 1

Пример частотной матрицы типа "слово—слово"

w/c	*the*	*quick*	*brown*	*fox*	*jumps*	*over*	*lazy*	*dog*
the	0	1	1	2	1	1	1	1
quick	1	0	1	1	1	0	0	0
brown	1	1	0	1	1	1	0	0
fox	2	1	1	0	1	1	0	0
jumps	1	1	1	1	0	1	1	0
over	1	0	1	1	1	0	1	1
lazy	1	0	0	0	1	1	0	1
dog	1	0	0	0	0	1	1	0

Таблица 2

Иллюстрация эвристики модели GloVe [1]

Статистика/*w*	*solid*	*gas*	*water*	*fashion*
$p(w\|ice)$	$1{,}9 \times 10^{-4}$	$6{,}6 \times 10^{-5}$	$3{,}0 \times 10^{-3}$	$1{,}7 \times 10^{-5}$
$p(w\|steam)$	$2{,}2 \times 10^{-5}$	$7{,}8 \times 10^{-4}$	$2{,}2 \times 10^{-3}$	$1{,}8 \times 10^{-5}$
$\frac{p(w\|ice)}{p(w\|steam)}$	8,9	$8{,}5 \times 10^{-2}$	1,36	0,96

На основе приведенных соображений авторами модели ставится следующая оптимизационная задача, в рамках решения которой строится векторное представление слов:

$$J(\Theta) = \sum_{w,c \in D} f(x_{wc}) \left(\langle \mathbf{v}_w, \mathbf{v}'_c \rangle + b_w + b'_w - \log x_{wc} \right)^2 \to \min_{\Theta},$$

где

$$\Theta = \left\{ (\mathbf{v}_w,\ \mathbf{v}'_w,\ b_w,\ b'_w) | w \in D \right\}, (\mathbf{v}_w,\ \mathbf{v}'_w \in \mathbb{R}^n,\ b_w,\ b'_w \in \mathbb{R}).$$

В качестве векторных представлений слов может использоваться каждый из следующих трех наборов векторов $\{\mathbf{v}_w\}$, $\{\mathbf{v}'_w\}$ или $\{\mathbf{v}_w + \mathbf{v}'_w\}$. В то же время авторы модели рекомендуют использовать третий набор, использование которого в ходе проведенных экспериментов показывает наилучшие результаты при решении задачи определения смысловой близости слов и задачи поиска аналогий.

В определении функционала J фигурирует весовая функция f. Эта функция является непрерывной и монотонно неубывающей. В нуле она принимает нулевое значение. С помощью этой функции штрафуются слишком большие значения счетчиков x_{wc}.

2. Задача классификации документов

В теории машинного обучения дается формальная постановка задачи классификации. В то же время следует отметить, что методы машинного обучения применимы только к математическим объектам, которыми тексты на естественном языке не являются. Поэтому, чтобы иметь возможность использовать понятия и методы машинного обучения применительно к задаче классификации документов, необходимо вначале заменить документы на их представления в рамках некоторой математической модели.

2.1. Математические модели текстов

В частотной матрице типа "слово — документ" каждому документу из рассматриваемого корпуса соответствует свой столбец. Такой столбец можно рассматривать как векторное представление документа. Исходная частотная матрица, к которой не применялась процедура взвешивания элементов, порождает модель документов под названием "мешок слов" BoW (*Bag of Words*) [15]. Частотная матрица, к элементам которой была применена функция взвешивания из семейства TF-IDF [24], порождает TF-IDF-модель документов.

Если задано некоторое векторное представление слов, то на его основе можно построить векторное представление документов. Наиболее распространенный подход заключается в суммировании векторов всех слов, встречающихся в документе. После этого полученная сумма усредняется (делится на число слов в документе). Если некоторое слово встречается в документе несколько раз, то при суммировании и усреднении учитывается каждое его появление в документе.

2.2. Машинное обучение

В общем виде задача классификации, рассматриваемая авторами настоящей статьи, имеет следующую постановку [9, 11]. Должны быть заданы множество объектов $\mathcal{X}$ и конечное множество классов $\mathcal{Y}$. В дальнейшем в качестве объектов классификации будут выступать документы, точнее их представления в соответствующей математической модели. Предполагается, что существует неизвестная функциональная зависимость между объектами и классами, о которой можно судить только по конечному множеству обучающих примеров $T = \{(x_i, y_i) | i = 1, ..., m\} \subset \mathcal{X} \times \mathcal{Y}$.

Решение рассматриваемой задачи классификации осуществляется в рамках некоторой модели обучения $M = (H, a)$. Подобная модель включает в себя множество гипотез H (функций вида $f : \mathcal{X} \to \mathcal{Y}$), среди которых ищется приближение к неизвестной функциональной зависимости, а также алгоритм a. Этот алгоритм для множества обучающих примеров T выбирает гипотезу $a(T) \in H$, которая трактуется как решение задачи классификации. Выбранную гипотезу называют классификатором, а выбор гипотезы интерпретируют как процесс обучения, в рамках которого строится классификатор.

Для оценки качества построенного классификатора используются различные числовые метрики. В общем случае подобная метрика имеет вид $\text{est}(T, f)$ и отражает соответствие гипотезы $f \in H$ множеству обучающих примеров T. Наиболее распространенной метрикой является точность (*accuracy*), которая представляет собой отношение числа успешно классифицированных примеров к общему числу примеров T. Несмотря на свою простоту и интуитивную понятность такая метрика имеет существенный недостаток. Этот недостаток может проявляться в случае, когда объекты неравномерно распределены по классам. В этой ситуации классификатор, оцененный как высокоточный, может неправильно классифицировать все объекты из некоторых классов.

Этого недостатка лишена метрика F_1, которая изначально применяется в случае бинарной классификации $\mathcal{Y} = \{0, 1\}$. В своем определении она использует две вспомогательных метрики:

$$F_1(T, f) = 2 \cdot \frac{\text{precision}(T, f) \cdot \text{recall}(T, f)}{\text{precision}(T, f) + \text{recall}(T, f)}.$$

Выделим три группы обучающих примеров.

Первая группа $TP(T, f) = \{(x, 1) \in T | f(x) = 1\}$.

Вторая группа $FP(T, f) = \{(x, 0) \in T | f(x) = 1\}$.

Третья группа $FN(T, f) = \{(x, 1) \in T | f(x) = 1\}$.

Тогда

$$\text{precision}(T, f) = \frac{|TP(T, f)|}{|TP(T, f)| + |FP(T, f)|}$$

и

$$\text{recall}(T, f) = \frac{|TP(T, f)|}{|TP(T, f)| + |FN(T, f)|}.$$

Метрика F_1 может быть обобщена на случай $|\mathcal{Y}| > 2$. С помощью анализируемой гипотезы f для каждого класса можно рассматривать отдельную бинарную задачу классификации (объект принадлежит этому классу или принадлежит любому другому классу). Следовательно, для каждого класса может быть получена своя F_1-оценка. Полученные оценки можно усреднить, или можно взять их взвешенную сумму.

Важным этапом решения задачи классификации является выбор подходящей модели обучения из ряда альтернатив либо, если модель обучения имеет настраиваемые параметры, для таких параметров требуется подобрать оптимальные значения. Возможны комбинации обозначенных вариантов. Проблема выбора модели тесно связана с необходимостью борьбы с двумя негативными явлениями, возникающими в процессе обучения, которые тесно связаны между собой. Первое явление носит название недообучения (*underfitting*). Оно возникает в ситуации, когда оценка $\text{est}(T, a(T))$ признается неудовлетворительной. Второе явление носит название переобучения (*overfitting*). Оно диагностируется в ситуации, когда построенный классификатор показывает хорошие результаты только на объектах из обучающих примеров. Наличие явления переобучения свидетельствует о том, что одной оценки $\text{est}(T, a(T))$ недостаточно, чтобы судить о качестве классификатора.

Множество обучающих примеров можно разбить на два непересекающихся множества $T = T_{\text{train}} \cup T_{\text{test}}$, называемых соответственно тренировочной и тестовой выборками. Тренировочная выборка, как правило, содержит 70 % обучающих примеров. Примеры только из обучающей выборки используются для построения классификатора. Оценка $\text{est}(T_{\text{train}}, a(T_{\text{train}}))$ показывает, имело ли место недообучение. Сравни-

вая между собой две оценки $\text{est}\left(T_{\text{train}}, a\left(T_{\text{train}}\right)\right)$ и $\text{est}\left(T_{\text{test}}, a\left(T_{\text{train}}\right)\right)$, можно сделать вывод о наличии переобучения.

3. Эксперименты

Настоящий раздел посвящен планированию и обсуждению результатов проведенных экспериментов над тестовыми наборами данных. На основе этих результатов делаются выводы о том, как влияет снижение размерности векторного представления слов, выполненное с применением метода главных компонент, на качество соответствующих моделей документов.

3.1. Тестовые наборы данных

В ходе проведения экспериментов было использовано три набора документов: movies (рецензии к кинофильмам), R8 (финансовые документы), twitter (сообщения из одноименной социальной сети). Опишем характеристики каждого из этих наборов.

Набор movies состоит из 44 012 документов, разбитых на шесть классов. Вектор (0,477, 0,28, 0,084, 0,079, 0,0451, 0,0266) описывает распределение документов по классам. Словарь состоит из 72 295 слов. Максимальный, минимальный и средний размер документа соответственно 698, 5 и 56 слов.

Набор R8 состоит из 7674 документов, разбитых на 8 классов. Вектор (0,51, 0,29, 0,048, 0,042, 0,038, 0,035, 0,018, 0,006) описывает распределение документов по классам. Словарь состоит из 17 387 слов. Максимальный, минимальный и средний размер документа соответственно 533, 4 и 64,5 слова.

Набор twitter состоит из 1 594 557 документов, разбитых на два класса. Вектор (0,5, 0,5) описывает распределение документов по классам. Словарь состоит из 35 738 слов. Максимальный, минимальный и средний размер документа соответственно 50, 1 и 12,75 слова.

Как можно видеть, из всех наборов тестовых данных сбалансированным является только набор twitter.

Описанные наборы данных были случайным образом разбиты на тренировочную и тестовые выборки, соответственно, 70 и 30 % обучающих примеров.

3.2. Построение векторных представлений

В ходе проведения экспериментов были использованы две модели векторных представлений слов — GloVe и Word2Vec (CBOW). В рамках этих моделей на основе тестовых наборов документов были построены векторные представления слов, имеющие размерность 50, 100, 300. Назовем эти векторные представления слов исходными. После этого исходные векторные представления слов были подвергнуты дополнительной обработке (постобработка).

Для каждого исходного векторного представления слов были вычислены главные компоненты. Переход к новому базису, образованному главными компонентами, порождает новое векторное представление слов. Будем именовать новое векторное представление слов, как имеющее тип PCA. Будем считать, что удаление из нового векторного представления слов первой (последней) координаты порождает векторное представление слов типа PCA-1 (PCA-*n*). Обратим внимание на то, что удаление первой (последней) координаты соответствует удалению проекции на первое (последнее) главное направление.

На основе исходных векторных представлений слов, а также векторных представлений слов типа PCA, PCA-1, PCA-*n* были построены векторные представления документов для тестовых наборов. В качестве векторного представления документа было взято среднее арифметическое векторов слов, входящих в его состав [11].

Дополнительно для всех документов из тестовых наборов были построены представления в модели BoW.

3.3. Результаты

В ходе проведения экспериментов были использованы две модели классификации, а именно — логистическая регрессия (простая модель) и случайный лес (сложная ансамблевая модель) [9]. Для этих целей были взяты реализации этих моделей из библиотеки Scikit-Learn[1]: класс LogisticRegression, реализующий модель логистической регрессии, и класс ExtraTreesClassifier, реализующий модель случайного леса. Каждый из этих классов имеет набор настраиваемых параметров.

В случае класса LogisticRegression задавался алгоритм решения соответствующей оптимизационной задачи (значения newton-cg, lbfgs, liblinear, sag, saga), параметр регуляризации (значения 1000, 100, 10, 1, 0,1) и начальное значение датчика псевдослучайных чисел (два значения). В случае класса ExtraTreesClassifier задавалось число деревьев (значения 50, 100, 200), максимальная глубина деревьев (значения 10, 20, 50, 100), минимальный размер выборки, которая может быть подвергнута разбиению (значения 2, 5, 10), и начальное значение датчика псевдослучайных чисел (два значения). Для каждой комбинации значений настраиваемых параметров и набора входных данных обучался и оценивался отдельный классификатор.

В табл. 3 приведены результаты экспериментов над классификаторами, построенными на основе элементарной модели документов BoW. По каждому набору тестовых данных и каждой модели классификации был построен, обучен и оценен целый набор классификаторов. Каждой допустимой комбинации значений настраиваемых параметров модели обучения соответствует свой построенный классификатор. Для каждого классификатора было вычислено значение метрики F_1. Максимальные значения этой метрики, достигнутые на тестовой выборке, представлены в табл. 3.

Аналогично устроены табл. 4 и 5, в которых приводятся результаты экспериментов над классификаторами, построенных на основе векторных представлений слов. Отличие состоит только в том, что в каждой ячейке указано три числовых значения, соответствующих исходным размерностям 50, 100 и 300 использованных векторных представлений

[1] https://scikit-learn.org

Таблица 3

Результаты классификации, соответствующие модели документов BoW

Набор	Логистическая регрессия	Случайный лес
movies	0,815951	0,306914
R8	0,940006	0,809663
twitter	0,795760	0,782860

слов. Таблица 4 содержит результаты, относящиеся к модели векторных представлений слов GloVe. В ней представлены результаты классификации для исходных векторных представлений слов и представлений типа PCA-1. Таблица 5 содержит результаты, относящиеся к модели векторных представлений слов Word2Vec (CBOW). В ней представлены результаты классификации для векторных представлений слов типа PCA-1 и PCA-n. Для обеих моделей Glove и Word2Vec (CBOW) были получены похожие результаты.

Сравнивая между собой результаты табл. 3 с результатами из табл. 4 и 5, можно сделать следующий вывод. Никаких принципиальных улучшений за счет использования векторных представлений слов в характеристиках построенных классификаторов добиться не удалось. Возможно, причина этого факта кроется в том, что была выбрана простейшая модель представления документа, использующая среднее арифметическое векторных представлений входящих в его состав слов. Возможно, переход к более сложной модели (например, Doc2Vec [27]) покажет принципиально другие результаты.

Обратимся к результатам табл. 4. Удаление проекции на первое главное направление не привело к улучшению точности классификации. Можно с уверенностью утверждать, что эффект, отмеченный в работе [10] для задач определения смысловой близости слов и поиска аналогий, для задачи классификации документов не наблюдается. Отмечено хоть и не очень большое, однако ожидаемое снижение точности.

Результаты табл. 5 позволяют ответить и на другой важный вопрос. Удаление проекций на главные направления ведет к ожидаемому снижению качества классификации. Возникает вопрос: "Какое направление выбрать, чтобы уменьшение точности классификации было бы минимальным?" Оказывается, традиционный подход, при котором удаляются проекции на последние главные направления, в большинстве случаев дает лучший результат.

Таблица 4

Результаты классификации, соответствующие моделям документов на основе векторного представления GloVe

Набор	Логистическая регрессия	Логистическая регрессия (PCA-1)	Случайный лес	Случайный лес (PCA-1)
movies	0,754832 0,765041 0,783604	0,751534 0,764436 0,784369	0,725637 0,699336 0,655621	0,714296 0,687812 0,624901
R8	0,965767 0,968497 0,972009	0,965341 0,966765 0,970631	0,960269 0,958534 0,957772	0,958854 0,950632 0,909754
twitter	0,742419 0,755872 0,766030	0,742051 0,755731 0,765964	0,755389 0,757037 0,754583	0,756333 0,760177 0,750803

Таблица 5

Результаты классификации, соответствующие моделям документов на основе векторного представления Word2Vec (CBOW)

Набор	Логистическая регрессия (PCA-1)	Логистическая регрессия (PCA-*n*)	Случайный лес (PCA-1)	Случайный лес (PCA-*n*)
movies	0,767103 0,773407 0,783248	0,781135 0,787616 0,794707	0,755885 0,740407 0,715982	0,771904 0,758773 0,735944
R8	0,966873 0,966374 0,965541	0,967717 0,967021 0,965835	0,937727 0,925723 0,901645	0,952296 0,941638 0,911607
twitter	0,745005 0,760794 0,773951	0,746636 0,761601 0,774420	0,739067 0,742377 0,734519	0,740025 0,742967 0,734316

Заключение

В рамках исследования, результаты которого представлены в данной работе, рассматривался ряд вопросов, посвященных использованию векторных представлений слов для решения задачи классификации текстовых документов. Как отмечалось ранее, модели векторных представлений слов являются популярным объектом теоретических исследований. Развитие этих моделей, в том числе переход от частотных моделей к предсказательным, стимулировалось потребностью снижения больших вычислительных затрат, возникающих при их практическом использовании. Эта проблема остается актуальной и в настоящее время. При решении практических задач часто приходится отказываться от самостоятельного построения векторного представления слов и пользоваться готовыми решениями. В результате, востребованным практикой становится решение задачи постобработки существующего векторного представления слов, например, снижения его размерности.

В последнее время появился ряд работ (например, [10]), в которых исследуется необычный подход к использованию метода главных компонент для снижения размерности векторных представлений слов. В рамках этого подхода удаляются проекции не на последние, а на первые главные компоненты. Оказывается, что применительно к задачам определения смысловой близости слов и поиска аналогий этот подход позволяет увеличить точность их решения. Экспериментальные исследования, результаты которых представлены в настоящей работе, показывают, что в решении задачи классификации документов подобного эффекта не наблюдается. Удаление проекций на главные направления приводят к снижению точности классификации. При этом традиционный подход, при котором удаляются проекции на последние главные направления, в большинстве случаев дает наилучший результат.

Список литературы

1. **Pennington J., Socher R., Manning C. D.** GloVe: Global Vectors for Word Representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). — 2014. — P. 1532—1544.
2. **Bullinaria J. A., Levy J. P.** Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study // Behavior Research Methods — 2007. — Vol. 39. — P. 510—526.
3. **Resnik P.** Using Information Content to Evaluate Semantic Similarity in a Taxonomy // Proceedings of the International Joint Conference for Artificial Intelligence (IJCAI-95). — 1995 — P. 448—453.
4. **Budanitsky A., Hirst G.** Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures // Workshop on WordNet and other lexical resources. — NAACL, 2001. — P. 29—34.
5. **Gentner D.** Structure-mapping: A theoretical framework for analogy // Cognitive Science. — 1983. — Vol. 7, No. 2. — P. 155—170.
6. **Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.** Distributed representations of words and phrases and their compositionality // Proceedings of the 26th International Conference on Neural Information Processing Systems — 2013. — Vol. 2. — P. 3111—3119.
7. **Mikolov T., Yih W., Zweig G.** Linguistic Regularities in Continuous SpaceWord Representations // HLT-NAACL. — 2013. — P. 746—751.
8. **Mikolov T., Chen K., Corrado G., Dean J.** Efficient Estimation of Word Representations in Vector Space // Computing Research Repository (CoRR). — 2013. — P. 1—12. URL: https: // arxiv.org/abs/1301.3781 (дата обращения 10.09.2020).
9. **Bishop C. M.** Pattern Recognition and Machine Learning. — Springer Science + Business Media LLC, 2006. — 738 p.
10. **Mu J., Bhat S., Viswanath P.** All-but-the-Top: Simple and Effective Post-processing for Word Representations // Computing Research Repository (CoRR) — 2018. — P. 1—25. URL: https: // arxiv.org/abs/1702.01417 (дата обращения 10.09.2020).
11. **Jurafsky D., Martin J.** Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. — Prentice Hall, 2nd Edition, 2008. — 1032 p.
12. **Шундеев А. С.** Об изменении размерности векторного представления текстовых данных // Программная инженерия. — 2019. — Т. 10, № 6. — С. 265—273.
13. **Shundeev A. S.** Towards document embeddings using dictionary transformation // Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, CEUR Workshop Proceedings (CEUR-WS.org) — 2019. — Vol. 2514. — P. 368—376.
14. **Shundeev A. S., Balakhnichev S. A., Zaslavskii D. D., Pekhterev S. I.** Document classification using word embeddings // Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, CEUR Workshop Proceedings (CEUR-WS.org) — 2019. Vol. 2514. — P. 377—388.
15. **Salton G., Wong A., Yang C. S.** A Vector Space Model for Automatic Indexing // Commun. ACM. — 1975. — Vol. 18, No. 11. — P. 613—620.
16. **Deerwester S. C., Dumais S. T., Landauer T. K., Furnas G. W., Harshman R. A.** Indexing by latent semantic analysis // Journal of the American Society for Information Science (JASIS). — 1990. — Vol. 41, No. 6. — P. 391—407.
17. **Turney P. D., Pantel P.** From Frequency to Meaning: Vector Space Models of Semantics // Journal of Artificial Intelligence Research. — 2010. — Vol. 37. — P. 141—188.
18. **Bengio Y., Ducharme R., Vincent P., Janvin C.** A Neural Probabilistic Language Model // Journal of Machine Learning Research. — 2003. — Vol. 3. — P. 1137—1155.
19. **Baroni M., Kruszewski G.** Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. — 2014. — Vol. 1. — P. 238—247.
20. **Levy O., Goldberg Y., Dagan I.** Improving Distributional Similarity with Lessons Learned from Word Embeddings // TACL. — 2015. — Vol. 3. — P. 211—225.
21. **Wittgenstein L.** Philosophical Investigations. — Blackwell. Translated by G. E. M. Anscombe, 1953. — 232 p.
22. **Harris Z.** Distributional structure // Word — 1954. — Vol. 10, No. 23. — P. 146—162.
23. **Firth J. R.** A synopsis of linguistic theory 1930—1955 / Studies in Linguistic Analysis. — Blackwell, Oxford, 1957. — P. 1—32.
24. **Sparck J. K.** A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation — 1972. — Vol. 28, № 1. — P. 11—21.
25. **Le Q., Mikolov T.** Distributed Representations of Sentences and Documents // Proceedings of the 31st International Conference on Machine Learning. — 2014. — Vol. 32, No. 2. — P. 1188—1196.
26. **Bojanowski P., Grave E., Joulin A., Mikolov T.** Enriching Word Vectors with Subword Information // Computing Research Repository (CoRR). — 2017. — P. 1—12. URL: https: // arxiv.org/abs/1607.04606 (дата обращения 10.09.2020).
27. **Wu L., Yen I. E. H., Xu K., Xu F., Balakrishnan A., Chen P., Ravikumar P., Witbrock M. J.** Word Mover's Embedding: From Word2Vec to Document Embedding // Computing Research Repository (CoRR). — 2018. — P. 1—15. URL: https: // arxiv.org/abs/1811.01713 (дата обращения 10.09.2020).

Reducing the Documents Embeddings Dimension Using Principal Component Analysis

A. S. Shundeev, alex.shundeev@gmail.com, **D. D. Zaslavskii**, zabaf@ya.ru, **S. I. Pekhterev**, stas-19000@mail.ru, Lomonosov Moscow State University, Moscow, 119192, Russian Federation

Corresponding author:

Shundeev Aleksandr S., Leading Researcher, Lomonosov Moscow State University, Moscow, 119192, Russian Federation
E-mail: alex.shundeev@gmail.com

Received on September 21, 2020
Accepted on October 20, 2020

In this paper, a number of issues related to the use of word embeddings for solving the problem of classifying text documents were considered. Models of word embeddings have been a popular object of theoretical research since the mid-1970. The development of these models, including the transition from count based models to predictive ones, was stimulated by the need to reduce the large computational costs arising from their practical use. This problem is still relevant today. When solving practical problems, one often has to abandon the independent construction of word embeddings and use ready-made solutions. As a result, the problem of post-processing of the existing word embeddings, for example, reducing its dimension, becomes urgent.

Recently, a number of works have appeared in which an unusual approach to using the principal component analysis to reduce the dimension of word embeddings has been investigated. In this approach, projections are removed not on the last, but on the first principal components. It turns out that in relation to the problems of determining the word similarity and word analogies, this approach can increase the accuracy of their solution. Experimental studies, the results of which are presented in this work, show that this effect is not observed in solving the problem of document classification. Removing projections to the first directions leads to a decrease in the classification accuracy. At the same time, the traditional approach, which removes projections to the last principal directions, in most cases gives the best result.

Keywords: *word embeddings, document embeddings, text classification, Word2Vec, GloVe, fastText, distributional hypothesis, post-processing, principal component analysis*

For citation:

Shundeev A. S., **Zaslavskii D. D.**, **Pekhterev S. I.** Reducing the Documents Embeddings Dimension Using Principal Component Analysis, *Programmnaya Ingeneria*, 2021, vol. 12, no. 1, pp. 48—57

DOI:10.17587/prin.12.48-57

References

1. **Pennington J., Socher R., Manning C. D.** GloVe: Global Vectors for Word Representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532—1544.

2. **Bullinaria J. A., Levy J. P.** Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study, *Behavior Research Methods*, 2007, vol. 39, pp. 510—526.

3. **Resnik P.** Using Information Content to Evaluate Semantic Similarity in a Taxonomy, *Proceedings of the International Joint Conference for Artificial Intelligence (IJCAI-95)*, 1995, pp. 448—453.

4. **Budanitsky A., Hirst G.** Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, *Workshop on WordNet and other lexical resources, NAACL*, 2001, pp. 29—34.

5. **Gentner D.** Structure-mapping: A theoretical framework for analogy, *Cognitive Science*, 1983, vol. 7, no. 2, pp. 155—170.

6. **Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.** Distributed representations of words and phrases and their compositionality, *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2013, vol. 2, pp. 3111—3119.

7. **Mikolov T., Yih W., Zweig G.** Linguistic Regularities in Continuous SpaceWord Representations, *HLT-NAACL*, 2013, pp. 746—751.

8. **Mikolov T., Chen K., Corrado G., Dean J.** Efficient Estimation of Word Representations in Vector, *Computing Research Repository (CoRR)*, 2013, pp. 1—12, available at: https://arxiv.org/abs/1301.3781.

9. **Bishop C. M.** *Pattern Recognition and Machine Learning*, Springer, Science + Business Media LLC, 2006, 738 p.

10. **Mu J., Bhat S., Viswanath P.** All-but-the-Top: Simple and Effective Post-processing for Word Representations, *Computing Research Repository (CoRR)*, 2018, pp. 1—25, available at: https://arxiv.org/abs/1702.01417.

11. **Jurafsky D., Martin J.** *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice Hall, 2nd Edition, 2008, 1032 p.

12. **Shundeev A. S.** On Changing the Dimension of the Document Embeddings, *Programmnaya Ingeneria*, 2019, vol. 10, no. 6, pp. 265—273 (in Russian).

13. **Shundeev A. S.** Towards document embeddings using dictionary transformation, *Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, CEUR Workshop Proceedings (CEUR-WS.org)*, 2019, vol. 2514, pp. 368—376.

14. **Shundeev A. S., Balakhnichev S. A., Zaslavskii D. D., Pekhterev S. I.** Document classification using word embeddings, *Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, CEUR Workshop Proceedings (CEUR-WS.org)*, 2019, vol. 2514, pp. 377—388.

15. **Salton G., Wong A., Yang C. S.** A Vector Space Model for Automatic Indexing, *Commun. ACM*, 1975, vol. 18, no. 11, pp. 613—620.

16. **Deerwester S. C., Dumais S. T., Landauer T. K., Furnas G. W., Harshman R. A.** Indexing by latent semantic analysis. *Journal of the American Society for Information Science (JASIS)*, 1990, vol. 41, no. 6, pp. 391—407.

17. **Turney P. D., Pantel P.** From Frequency to Meaning: Vector Space Models of Semantics, *Journal of Artificial Intelligence Research*, 2010, vol. 37, pp. 141—188.

18. **Bengio Y., Ducharme R., Vincent P., Janvin C.** A Neural Probabilistic Language Model, *Journal of Machine Learning Research*, 2003, vol. 3, pp. 1137—1155.

19. **Baroni M., Kruszewski G.** Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, *ACL,* 2014, vol. 1, pp. 238—247.

20. **Levy O., Goldberg Y., Dagan I.** Improving Distributional Similarity with Lessons Learned from Word Embeddings, *TACL*, 2015, vol. 3, pp. 211—225.

21. **Wittgenstein L.** *Philosophical Investigations*, Blackwell. Translated by G. E. M. Anscombe, 1953, 232 p.

22. **Harris Z.** Distributional structure, *Word*, 1954 vol. 10, no. 23, pp. 146—162.

23. **Firth J. R.** A synopsis of linguistic theory 1930—1955. *In Studies in Linguistic Analysis*, Blackwell, Oxford, 1957, pp. 1—32.

24. **Sparck J. K.** A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 1972, vol. 28, no. 1, pp. 11—21.

25. **Le Q., Mikolov T.** Distributed Representations of Sentences and Documents, *Proceedings of the 31st International Conference on Machine Learning*, 2014, vol. 32, no. 2, pp. 1188—1196.

26. **Bojanowski P., Grave E., Joulin A., Mikolov T.** Enriching Word Vectors with Subword Information, *Computing Research Repository (CoRR)*, 2017, pp. 1—12, available at: https://arxiv.org/abs/1607.04606.

27. **Wu L., Yen I. E. H., Xu K., Xu F., Balakrishnan A., Chen P., Ravikumar P., Witbrock M. J.** Word Mover's Embedding: From Word2Vec to Document Embedding. *Computing Research Repository (CoRR)*, 2018, pp. 1—15, available at: https://arxiv.org/abs/1811.01713.

ИНФОРМАЦИЯ

22—23 апреля 2021 г. состоится
Отраслевая научно-техническая конференция
приборостроительных организаций ГК "РОСКОСМОС"

"ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ И ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ",

посвященная 55-й годовщине образования АО "НПО ИТ"

Конференция проводится АО "Научно-производственное объединение измерительной техники" (АО "НПО ИТ") при поддержке государственной корпорации по космической деятельности "Роскосмос" и АО "Российские космические системы".

Заседания конференции будут проходить в АО "НПО ИТ" по адресу: Московская область, г. Королёв, ул. Пионерская, дом 2.

Тематика научно-технической конференции

1. Состояние и основные направления развития телеметрических и информационно-управляющих систем
2. Новое поколение интеллектуальной датчико-преобразующей и управляющей аппаратуры. Теория и принципы построения
3. Информационно-измерительное обеспечение летных испытаний
4. Волоконно-оптические гироскопы и приборы на их основе
5. Перспективы развития средств выведения и наземной космической инфраструктуры
6. Новые конструкции, технологии и материалы. Качество производства продукции приборостроительной отрасли
7. Двойные технологии. Техническое перевооружение приборостроительных организаций "Роскосмоса"

Адрес оргкомитета:
Российская Федерация, 141074, Московская область, г. Королёв, ул. Пионерская, дом 2,
Акционерное общество "Научно-производственное объединение измерительной техники"
(АО "НПО ИТ")

Контактные телефоны:
+7(499)750-40-50, доб. 11-66 Ачкасов Виталий Анатольевич
+7(499)750-40-50, доб. 10-01 Сабко Владимир Леонидович
+7(499)750-40-50, доб. 10-29 Вербицкая Карина Викторовна

Указатель статей, опубликованных в журнале "Программная инженерия" в 2020 г.

31 мая — 4 июня 2021 г., ДГТУ, Ростов-на-Дону, Россия

Международная научная мультиконференция

КИБЕР-ФИЗИЧЕСКИЕ СИСТЕМЫ: ПРОЕКТИРОВАНИЕ И МОДЕЛИРОВАНИЕ
CYBER-PHYSICAL SYSTEMS DESIGN AND MODELLING
(CYBERPHY-2021)
(SCOPUS, SPRINGER)

Секции

1. Cyber-Physical Systems: digital technologies and applications
   (Кибер-физические системы: цифровые технологии и приложения)
2. Cyber-physical systems: design and application for Industry 4.0
   (Кибер-физические системы: проектирование и применение для Индустрии 4.0)
3. Cyber-Physical Systems: Modelling and Intelligent Control
   (Кибер-физические системы: моделирование и интеллектуальное управление)
4. Society 5.0: Cyberspace for advanced human-centered society
   (Общество 5.0: киберпространство для развитого общества, ориентированного на человека)

XXXIV Международная научная конференция
МАТЕМАТИЧЕСКИЕ МЕТОДЫ В ТЕХНИКЕ И ТЕХНОЛОГИЯХ — ММТТ-34
(РИНЦ, DOI)

Секции

1. Качественные и численные методы исследования дифференциальных и интегральных уравнений
2. Оптимизация, автоматизация и оптимальное управление технологическими процессами
3. Математическое моделирование технологических и социальных процессов
4. Математическое моделирование и оптимизация в задачах САПР, аддитивных технологий, цифрового производства
5. Математические методы в задачах радиотехники, радиоэлектроники и телекоммуникаций, геоинформатики, авионики и космонавтики
6. Математические методы и интеллектуальные системы в робототехнике и мехатронике
7. Математические методы в медицине, биотехнологии и экологии
8. Математические методы в экономике и гуманитарных науках
9. Информационные и интеллектуальные технологии в технике и образовании
10. Математические и инструментальные методы технологий Индустрии 4.0
11. Обсуждение квалификационных работ

Подача заявок на участие с 15 декабря 2020 г.

Подробная информация о конференции и условиях участия в ней размещается на сайте http://mmtt.sstu.ru/

ООО "Издательство "Новые технологии". 107076, Москва, Стромынский пер., 4
Технический редактор *Е. М. Патрушева*. Корректор *Е. В. Комиссарова*

Сдано в набор 13.11.2020 г. Подписано в печать 24.12.2020 г. Формат 60×88 1/8. Заказ PI121
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: **www.aov.ru**

Рисунки к статье **Д. Д. Руховича**

«ИТЕРАТИВНЫЙ МЕТОД ОБНАРУЖЕНИЯ ОБЪЕКТОВ»

Базовый метод. Точность: 74,28 %

а)

IterDet, 1 итерация. Точность: 74,27 %
IterDet, 2 итерации. Точность: 79,59 %

б)

Рис. 1. Сравнение результатов Faster RCNN (*а*) и Faster RCNN-IterDet (*б*) на тестовом изображении из набора CrowdHuman. Рамки объектов, найденных на первой и второй итерациях, показаны зеленым и желтым цветом соответственно. В правом верхнем углу показан сложный случай двух сильно перекрывающихся объектов, с которым Faster RCNN-IterDet успешно справляется, в то время как Faster RCNN выдает ошибочное предсказание

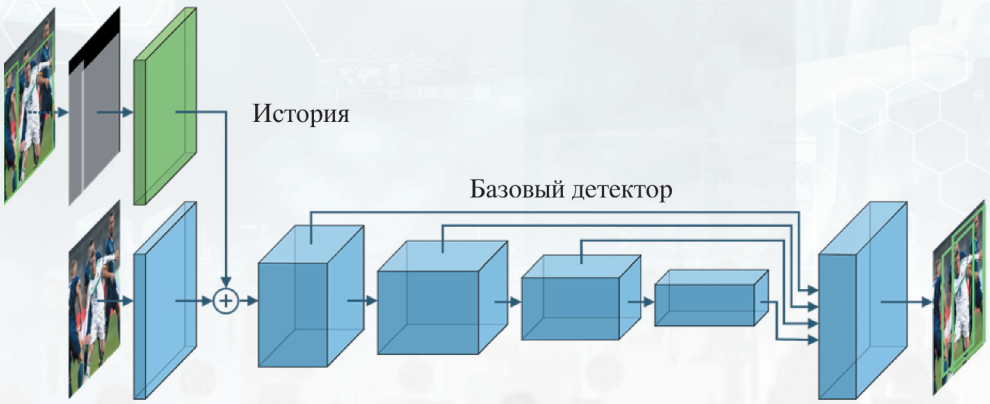

Рис. 2. Архитектура итеративного метода обнаружения объектов

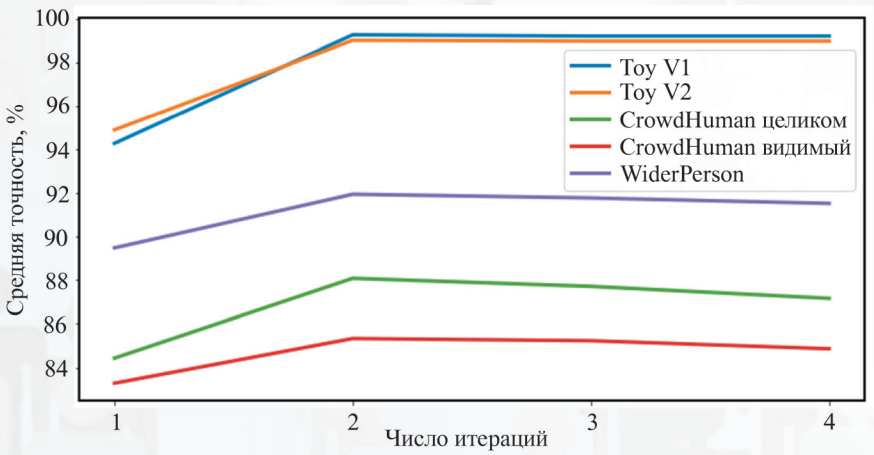

Рис. 3. Сравнение средней точности при различном числе итераций для IterDet с Faster RCNN-IterDet

Рисунки к статье **Д. Д. Руховича**
«ИТЕРАТИВНЫЙ МЕТОД ОБНАРУЖЕНИЯ ОБЪЕКТОВ»

а)

б)

в)

Рис. 4. Результаты обнаружения объектов с использованием итеративного метода на наборах Toy V1 и Toy V2 (*а*), CrowdHuman (*б*) и WiderPerson (*в*)

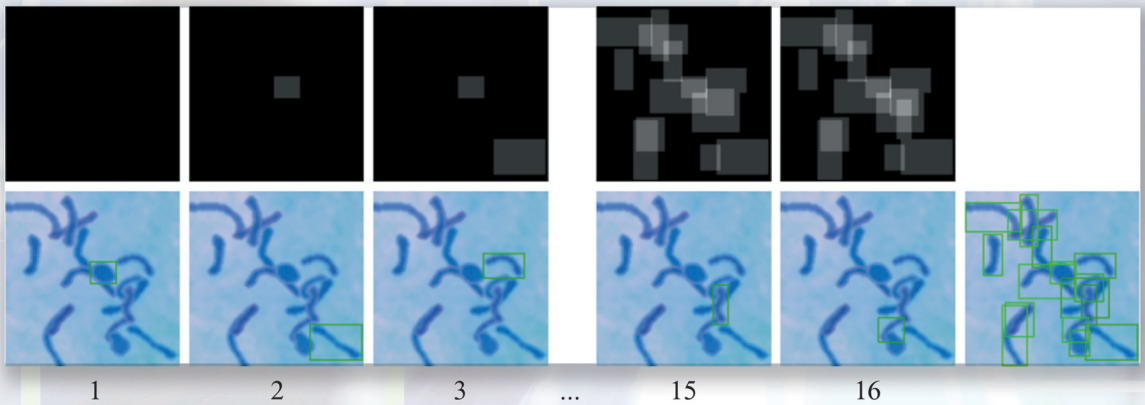

Рис. 5. Визуализация предсказаний, полученных с помощью метода IterDet, на изображении из набора Toy V2

ISSN 2220-3397. Программная инженерия. 2021. Том 12. № 1. 1 – 60. Подписной индекс по Объединенному каталогу "Пресса России" – 22765