

Программная инженерия



Пр **6**
ИН **2021**
Том 12

Рисунок к статье А. А. Титовой, В. А. Роганова, Г. А. Лукьянченко, С. Г. Елизарова
 «ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ТРАНЗАКЦИЙ В БЛОКЧЕЙН-СЕТЯХ,
 ИСПОЛЬЗУЮЩИХ АЛГОРИТМ CRYPTONIGHT В КАЧЕСТВЕ PROOF OF WORK
 НА БАЗЕ СПЕЦИАЛИЗИРОВАННОГО МНОГОЯДЕРНОГО ПРОЦЕССОРА MALT»

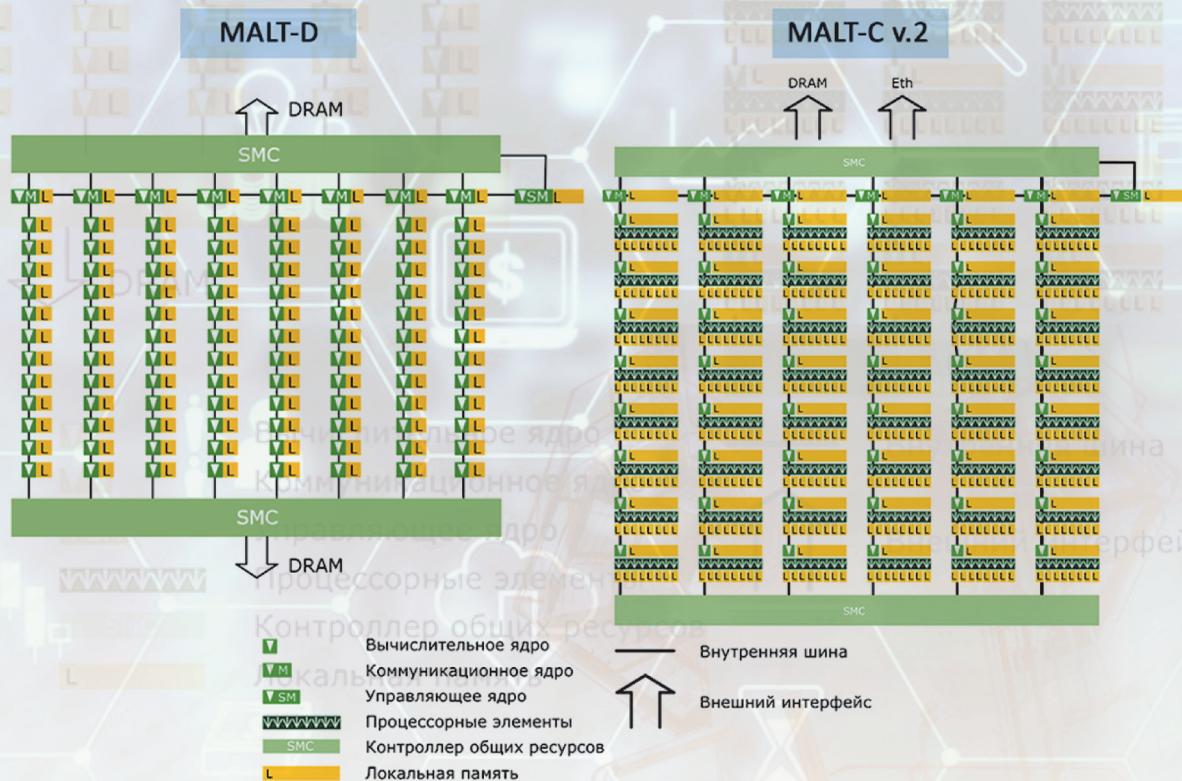


Рис. 1. Схемы процессоров MALT и расположения процессорных элементов SIMD-сопроцессора

Рисунок к статье А. С. Акопова, Н. К. Хачатряна
 «ИМИТАЦИОННАЯ МОДЕЛЬ НЕФТЕПЕРЕРАБАТЫВАЮЩЕГО ПРЕДПРИЯТИЯ
 СО СЛОЖНОЙ СХЕМОЙ ВЗАИМОДЕЙСТВИЯ ОБОРУДОВАНИЯ»

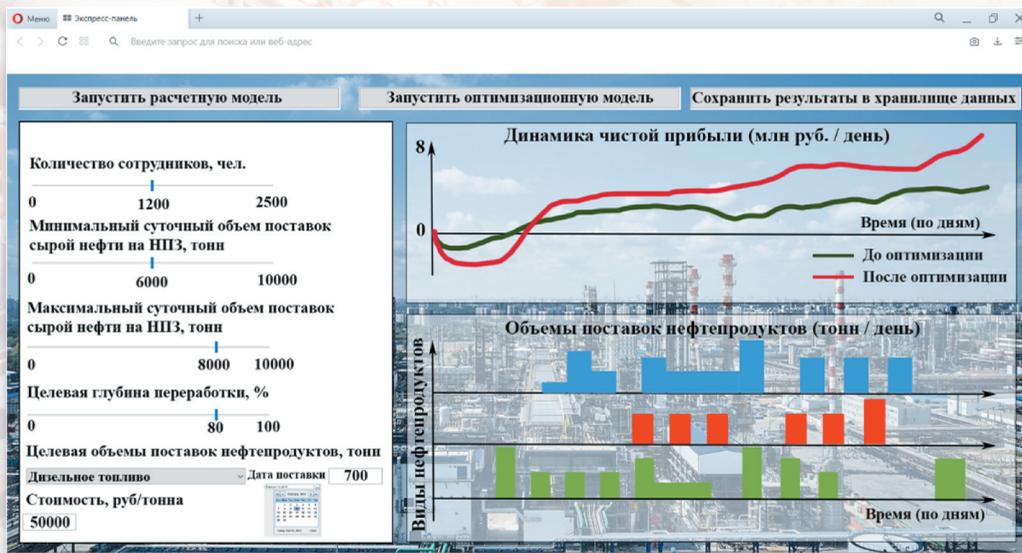


Рис. 4. Панель управления имитационной моделью НПЗ

Программная инженерия

Том 12
№ 6
2021
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

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

- Лунев К. В.** Алгоритм автоматизированной генерации обучающей выборки для решения задачи выявления семантической близости между парой ключевых слов методами машинного обучения 283
- Титова А. А., Роганов В. А., Лукьянченко Г. А., Елизаров С. Г.** Повышение эффективности транзакций в блокчейн-сетях, использующих алгоритм Cryptonight в качестве Proof of Work на базе специализированного многоядерного процессора MALT 295
- Бессонов О. А., Харченко Н. А.** Программная платформа для суперкомпьютерного моделирования задач аэротермодинамики 302
- Пащенко Д. С.** Российский опыт организации полностью удаленной разработки программного обеспечения: отраслевое исследование 2021 г. . . 311
- Акопов А. С., Хачатрян Н. К.** Имитационная модель нефтеперерабатывающего предприятия со сложной схемой взаимодействия оборудования 319
- Туровский Я. А., Борзунов С. В., Вахтин А. А.** Повышение частотного разрешения в задачах исследования ЭЭГ на основе ковариационного анализа 329

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.

© Издательство "Новые технологии", "Программная инженерия", 2021

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 12

N 6

2021

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

- Lunev K. V.** Algorithm for Automated Generation of a Training Sample for Solving the Problem of Determining Semantic Similarity between a Pair of Keywords using Machine Learning Methods 283
- Titova A. A., Roganov V. A., Lukyanchenko G. A., Elizarov S. G.** Transaction Accelerator for Blockchain Networks Based on Cryptonight Algorithm using Specialized Multicore Processor MALT . . 295
- Bessonov O. A., Kharchenko N. A.** Software Platform for Super-computer Modeling of Aerothermodynamics Problems 302
- Pashchenko D. S.** Russian Experience in Organizing Fully Remote Software Development: an Industry Study of 2021 311
- Akopov A. S., Khachatryan N. K.** Simulation Model of an Oil Refinery with a Complex Scheme of Equipment Interaction 319
- Turovsky Ya. A., Borzunov S. V., Vahtin A. A.** Increasing Frequency Resolution in EEG Studies based on Covariation Analysis 329

К. В. Лунев, аспирант, kirilllunev@gmail.com, Механико-математический факультет, Институт механики, МГУ имени М. В. Ломоносова

Алгоритм автоматизированной генерации обучающей выборки для решения задачи выявления семантической близости между парой ключевых слов методами машинного обучения

В настоящее время машинное обучение является эффективным подходом к решению множества задач информационно-аналитических систем. Для использования таких подходов требуется обучающее множество примеров. Сбор обучающего набора данных обычно является трудоемким процессом. Как правило, для его реализации требуется участие нескольких экспертов в предметной области, для которой собирается тренировочное множество. Более того, для некоторых задач, в число которых входит и задача определения семантической близости пар ключевых слов, является сложным даже правильно составить инструкцию для экспертов для адекватной оценки тестируемых примеров. Причина таких затруднений заключается в том, что семантическая близость — субъективная величина, которая сильно зависит от области применения, контекста, человека и задачи. В статье представлены результаты исследований на направлении поиска моделей, алгоритмов и программных средств для автоматизированного формирования объектов обучающей выборки в задаче определения семантической близости пары слов. Кроме того, модели, построенные по автоматизированной обучающей выборке, позволяют решать не только задачу определения семантической близости, но и произвольную задачу классификации ребер в графе. Методы, использованные в работе, основаны на теоретико-графовых алгоритмах.

Ключевые слова: семантическая близость, обработка естественного языка, алгоритмы на графах, теория графов, машинное обучение

Введение

Исследования, результаты которых представлены в настоящей работе, затрагивают важные и востребованные практикой задачи интеллектуального анализа объектов информационно-аналитической наукометрической системы.

В связи с высокой эффективностью методов машинного обучения с учителем в области определения уровня смысловой близости возникает необходимость сбора обучающей информации для подлежащих обучению моделей. В настоящей работе описан новый алгоритм автоматического формирования обучающей выборки для проведения обучения. Важность такого алгоритма заключается в том, что он избавляет от необходимости ручной разметки данных, которая обычно является ресурсозатратной.

Основной областью применения описанных далее методов является задача определения семантической близости пары ключевых слов. Особое значение имеет тот факт, что разработанные подходы в равной степени применимы к любым задачам, данные в которых представимы в графовом виде. Примером

такой задачи может служить задача классификации постов в социальной сети, где каждый пост представим в виде вершины графа, а ребра отражают различные связи между парой постов (например, факт причастности двух постов одному сообществу или одному автору).

Задача, решение которой представлено в статье, имеет следующую математическую постановку. Дано множество объектов системы D , множество ключевых слов W и множество наборов ключевых слов T . Каждый элемент $d_i \in D$, $i = 1, \dots, N$ представлен набором ключевых слов $t_i \in D$, состоящим из k_i ключевых слов из множества W :

$$d_i \rightarrow t_i = (w_{i_1}, \dots, w_{i_{k_i}}).$$

Множество T состоит из уникальных наборов ключевых слов. Наборы ключевых слов совпадают, если существует перестановка, отождествляющая эту пару наборов. В рамках данной работы будет представлена модель определения семантической близости, а в ее рамках — такая функция близости $f: W \times W \rightarrow [0, 1]$, большие значения которой озна-

чают сильный уровень смысловой близости между ключевыми словами.

В целях анализа эффективности существующих решений автором были проведены библиографические исследования, результаты которых представлены далее. В зависимости от числа доступных размеченных данных выделяется несколько различных типов задач машинного обучения. Далее представлен обзор этих типов.

1. Обучение по размеченным данным. В этом случае имеется достаточный набор обучающих примеров для прямого применения моделей машинного обучения.

2. Обучение с частично размеченными данными. В таком типе задач выборка содержит как размеченные, так и неразмеченные данные. В большинстве задач неразмеченная часть выборки значительно превосходит размеченную. При решении применяются методы, позволяющие эффективно использовать неразмеченную часть данных. Такие подходы позволяют улучшить качество предсказания по сравнению с моделями, использующими исключительно размеченные данные при обучении.

3. Обучение по неразмеченным данным. Весь объем данных является неразмеченным. Наиболее популярной задачей данного типа является задача кластеризации, в которой необходимо объединить объекты в группы (кластеры), исходя из близости этих объектов в признаковом пространстве.

4. Самообучение. Как и в предыдущем случае, изначально все данные являются неразмеченными. Первым шагом в решении задачи является разработка метода, позволяющего искусственно разметить данные, поставив каждому или некоторым объектам псевдо-метку, после чего задача решается методами из п. 1 и 2. Именно к этому типу задач относятся подходы, используемые в настоящей работе.

Модели самообучения широко распространены в задачах обработки естественного языка. Примерами таких моделей служат модели Word2Vec [1], GloVe [2] и StarSpace [3]. Указанные модели представляют собой нейронные сети, на вход которых подаются большие корпуса неразмеченных текстовых данных. Задачей обучения является построение такого векторного представления для текущего слова (*word embeddings*), которое максимально точно способно предсказать рядом стоящие в тексте слова. В этот момент в задаче появляются искусственно созданные метки: для данного слова X , слово Y является положительным примером, если оно встречается в тексте рядом со словом X , и отрицательным в ином случае. Обученная модель строит векторное пространство, обладающее рядом полезных свойств, которые в наше время широко используются для решения многих задач естественного языка, связанных с семантической информацией. Одно из таких свойств — семантическая близость понятий, векторные представления которых похожи. Таким образом, искусственно созданная обучающая выборка помогает решать реальную задачу определения семантической близости между словами естественного языка.

Данные методы являются очень эффективными, однако требуют огромные наборы данных для обучения моделей. Это обстоятельство делает их неприменимыми к задачам, в которых полные тексты документов недоступны. К их числу принадлежит задача определения семантической близости пары ключевых слов научных публикаций. Эти методы зачастую определяют также контекстную близость, по определению которой два слова близки, если они встречаются в похожих контекстах. Во многих практических задачах подобного эффекта использования метрики близости хочется избежать, поскольку, например, слова "математика" и "физика" могут встречаться в одних и тех же контекстах, но как пара ключевых слов для научных публикаций эти слова явно не являются семантически близкими.

Несмотря на высокое качество определения семантической близости моделей, использование полнотекстовой информации существенно ограничивает область применения данных методов, поскольку для многих прикладных задач не имеется достаточно количества текстовой информации. Возникает трудность при работе в узкоспециальных областях: модели, обученные на корпусах общего назначения, не могут улавливать особенности таких областей. Использование текстовых данных из рассматриваемой области для обучения ведет к неправильной настройке параметров модели и недообучению по причине недостатка этих самых данных. Это приводит к низкому уровню качества моделей.

Различные методы самообучения используются также и в задачах из других областей. Примером такой области служит обработка изображений. В задачах этого типа самообучение также используется для построения представлений (теперь уже для изображений, а не для слов), после чего представления используются для решения основной задачи (например, классификации изображений). Самообучение в этом случае подразумевает алгоритм создания обучающей выборки по набору неразмеченных изображений. Например, в работе [4] авторы искусственно поворачивают изображение на определенный угол и модель учится определять этот угол. В работе [5] применяется другой метод создания искусственной выборки. Из каждого изображения выделяется некоторый участок (называемый центральным), а также участок, находящийся около центрального (соседний). Задачей является предсказать то, как расположен соседний участок относительно центрального (выше, ниже, левее или правее).

Задачи, которые описываются данными графовой природы, также могут быть решены методами самообучения. Например, авторами работ [6, 7] разработан метод построения представления для вершины графа с помощью нейронных сетей. Для обучения таких моделей вводится искусственная разметка данных, в которой пара вершин является положительным примером, если вершины расположены близко друг к другу в графе, и отрицательным — в другом случае. Недостатком указанных подходов, как и в случае с описанными ранее моделями, является необходимость больших по размерам объемов входных данных.

В настоящей работе описаны подходы, позволяющие эффективно задействовать методы машинного обучения с учителем даже на выборках небольшого размера. Для этого автором разработан способ автоматизированного сбора обучающей выборки на основе графового представления данных. В рамках задачи определения семантической близости пары слов такой подход позволяет избавиться от трудоемкой задачи сбора обучающих примеров. Сложность этого процесса заключается, во-первых, в необходимости больших человеческих ресурсов для ручного сбора необходимой информации, а во-вторых, в потребности в объективных критериях для разметки.

В следующих двух разделах представлены способы формирования обучающих примеров для обучения: эвристический и автоматизированный. Эвристический способ использует простые идеи определения близких по смыслу ключевых слов и применяется в большей мере для валидации результатов автоматизированного способа. Автоматизированный подход является основным объектом исследования в рамках данной работы. Этот подход призван решить трудоемкую задачу сбора обучающих примеров.

Следующие разделы посвящены признаковому описанию модели и тестовым испытаниям программных реализаций описанных алгоритмов. После чего представлены выводы и дальнейшие планы по развитию исследуемых подходов.

Эвристические методы сбора обучающей выборки

С позиции методологии различают два типа эвристических алгоритмов сбора обучающей выборки. Первый из них заключается в использовании некоторых внешних словарей и дальнейшей фильтрации этих словарей по тем словам, которые присутствуют в информационной системе. Второй способ сначала использует некоторый генеративный алгоритм, а именно алгоритм, который для заданного слова генерирует различные слова-кандидаты, потенциально близкие по смыслу к заданному. В качестве такого алгоритма может выступать, например, алгоритм, который в ходе работы для ключевого слова выбирает все слова, которые встретились с заданным внутри одного набора. Далее полученное множество фильтруется с помощью некоторого фильтрующего алгоритма.

В результате работы программной реализации такого алгоритма определяются пары ключевых слов, которые принимают за семантически близкие. Этот алгоритм должен обладать высокой точностью, хотя и возможно низкой полнотой определения семантической близости. Другими словами, пары ключевых слов, объявленные семантически близкими, должны действительно являться похожими с вероятностью, близкой к 100 %. Однако следует отметить, что при этом число таких пар может быть относительно невелико.

Примером такого алгоритма может быть алгоритм, считающий расстояние Левенштейна. Если редакторское расстояние не превосходит 1, то слова

имеют очень схожее написание и весьма вероятно, что они близки по смыслу. Такая процедура фильтрует значительную часть пар. Однако те пары, которые прошли фильтрацию, почти всегда имеют высокую степень смысловой близости. Таким образом, каждый такой фильтр получает примеры близких по смыслу пар слов с высокой точностью, но низкой полнотой. Собранные с помощью такого подхода пары ключевых слов являются положительными примерами обучающей выборки для дальнейшего обучения.

В рамках исследований автором были разработаны перечисленные далее эвристические методы сбора обучающей выборки.

- Поиск простых аббревиатур. Ключевые слова системы разделяются на понятия, содержащие ровно одно слово — аббревиатуры, и понятия, содержащие более одного слова — расшифровки аббревиатур. Далее для каждого слова из множества аббревиатур и каждого слова из множества расшифровок проверяется, действительно ли данная расшифровка является расшифровкой для данной аббревиатуры. Другими словами, проверяется выполнение условия, что существуют такие префиксы слов расшифровки, которые могут полностью покрыть аббревиатуру. Если соответствие установлено, то пара аббревиатура—расшифровка добавляется в список кандидатов. Далее к парам-кандидатам приписываются частоты входящих в них ключевых слов. Для каждой аббревиатуры берется не более 5 наиболее частотных вариантов расшифровок. Пары, прошедшие данную фильтрацию, формируют окончательное обучающее множество аббревиатур.

- Поиск скобочных аббревиатур. Иногда при написании ключевых слов-аббревиатур в скобках указывается правильная расшифровка для данной аббревиатуры. Такие действия позволяют собрать дополнительные пары аббревиатура—расшифровка для обогащения множества обучающих примеров.

- Поиск разных форм одного слова. С помощью пакета обработки естественного языка NLTK для каждого слова в системе рассматриваются различные его формы. Если одна из форм также присутствует в множестве всех ключевых слов, то пара слово—форма добавляется в обучающее множество. Отметим, что методы морфологического анализа в настоящее время очень развиты и обычно применяются для предобработки текстовых данных до интеллектуального анализа. В рамках данной работы предварительный морфологический анализ не был проведен. Причина заключается именно в том, что словоформы являются хорошими примерами семантически близких слов. Если же на этапе предобработки привести все слова к начальной форме, то это уменьшит размер эвристической выборки и усложнит процесс валидации рассматриваемых в данной работе моделей.

- Поиск похожих по написанию слов. Рассматриваются все пары слов, каждое из которых состоит минимум из 7 символов и расстояние Левенштейна между которыми равно 1. При этом накладывается ограничение на длину слов. Ограничение на размер слова необходимо, поскольку изменение одной буквы

в коротком слове с большей вероятностью изменит смысл этого слова. Примером такой пары может служить пара "день"—"лень". Эти пары добавляются в обучающее множество.

- Поиск переводов с одного языка на другой. Данный метод использует API сервиса Яндекс.Переводчик и собирает варианты переводов ключевых слов с русского языка на английский.

- Поиск синонимов в тезаурусе WordNet. Были использованы пары синонимов и пары гипоним—гипероним. Кроме поиска по англоязычным синсетам был использован двухступенчатый подход определения синонимов: русские слова переводились средствами сервиса Яндекс.Переводчик на английский язык, а затем поиск проводился уже по английским версиям слов. Для увеличения уровня точности использовались слова, имеющие ровно один перевод с русского на английский.

- Использование открытых источников синонимов. Из словарей синонимов, доступных в сети Интернет, выделяются те пары, слова которых присутствуют в множестве ключевых слов.

Отмеченные методы сбора данных позволяют собрать положительные примеры для обучения, т. е. примеры пар ключевых слов, являющиеся семантически похожими. Однако для правильного обучения модели необходимы также и отрицательные примеры. С помощью небольших модификаций эвристических алгоритмов сбора положительных примеров можно получить алгоритмы для сбора отрицательных примеров. На этом направлении использованы следующие идеи.

- Поиск антонимов в тезаурусе WordNet. Аналогично поиску синонимов был проведен поиск антонимов в базе WordNet. Также были использованы пары слов, расстояние в дереве WordNet между которыми превышает 2.

- Неправильные расшифровки для аббревиатур. Рассматриваются те аббревиатуры, для которых были найдены правильные расшифровки. Этим аббревиатурам ставятся в пару случайные многословные ключевые слова, которые точно не являются правильными расшифровками.

- Использование открытых источников антонимов.

- Случайные пары ключевых слов. Для каждого ключевого слова, для которого в выборке присутствуют положительные примеры, были взяты случайные ключевые слова в пару. На одно слово было сгенерировано не более, чем 10 пар. Данный метод был использован по следующим соображениям. Предполагается, что для каждого ключевого слова существует константное число близких по смыслу ключевых слов. То есть если в достаточно большой информационной системе начинает расти число ключевых слов, то число слов, похожих на данное слово, не будет расти линейно по числу уникальных слов системы. В то же время число пар растет квадратично, а число пар для заданного слова — линейно, а это означает, что если взять для заданного слова в пару случайное слово, то эта пара не будет связана семантически. При этом важным является

подбор доли случайных пар. Если эта доля мала, то в выборке не будет достаточного разнообразия отрицательных примеров. Если же доля слишком велика, то случайные пары будут вносить слишком большой вклад в обучение модели. Это не является положительным фактором, потому что случайные отрицательные пары не настолько качественны, как, например, антонимы из словаря. Оба этих случая приводят к ухудшению качества модели определения семантической близости.

В результате работы программных реализаций перечисленных выше алгоритмов было собрано 234 974 положительных и 1 175 610 отрицательных примеров.

Обучение моделей на эвристически подобранных выборках имеет представленные далее заметные недостатки.

- Смещение в данных. Для наилучшего обучения классификатора необходимо, чтобы объекты обучающей выборки выбирались из распределения тех объектов, на которых классификатор будет работать в реальной системе. Рассмотрим данную особенность на следующем тривиальном примере. Допустим, что все пары ключевых слов системы делятся на аббревиатуры (т. е. пара аббревиатура—расшифровка аббревиатуры) и переводы (слово на русском языке — его перевод на английский). Если аббревиатур в системе 10 %, то и в обучающем подмножестве аббревиатур должно быть 10 %. Если же аббревиатур при обучении будет значительно больше (например, 70 %), то модель начнет усиленно использовать те факторы, которые повышают ее качество на аббревиатурной части выборки, потому что это лучшим образом оптимизирует функцию потерь. Однако в момент применения модели к ней на вход будет поступать только 10 % аббревиатур и 90 % переводов и очень вероятно, что факторы, используемые моделью, не будут оптимальны для классификации переводов, что ухудшит качество реальных решаемых данным классификатором задач. Таким образом, эвристически собранные выборки не являются репрезентативными для данных системы, в которой будут использоваться. Кроме этого, со временем распределение различных типов в системе может изменяться, что еще сильнее усугубит проблему смещения в данных при обучении.

Другой недостаток смещения заключается в том, что если перегрузить обучающую выборку примерами неправильных переводов, то это негативно отразится на предсказанных уровнях близости во время этапа применения модели. Как следствие, это понизит средний уровень близости между переводами в системе. Отмеченное понижение приведет к тому, что если для данного ключевого слова (MSU) есть и правильная аббревиатура (Moscow State University), и правильный перевод (МГУ), то вследствие рассмотренной манипуляции с обучающей выборкой значимость перевода будет понижена. В конечном счете может оказаться, что согласно модели, все аббревиатуры лучше всех переводов. Такое поведение не очевидно и не основывается на реальной работе информационной системы. Такой сценарий возможен, поскольку положительные и отрицательные

примеры берутся из разных источников. Поэтому степень покрытия ими всего разнообразия пар ключевых слов, а также возможность контролировать нужную долю положительных примеров остается под вопросом.

- Модель настраивается не под конкретную задачу и не под конкретные данные системы. Если, например, слова "объединять" и "интегрировать" попали в один внешний словарь синонимов общего назначения, то внутри информационной системы, направленной на изучение технических наук, эти слова определяют две различные математические операции. Может также случиться, что в узкоспециальных областях понятия могут являться слишком общими по смыслу, чтобы быть похожими. Примером такой пары могут быть "ураган" и "тайфун". Для пользователя, например, социальной сети эти слова действительно похожи. Однако в рамках системы, изучающей различные природные явления, слова имеют важные смысловые отличия. Чем большей спецификой обладает тематика системы, тем четче прослеживаются различия между словами, которые с точки зрения словарей общей направленности очень похожи.

- Сложность процесса сбора обучающей выборки. Для каждой системы нужны свои наборы эвристики, что затрудняет внедрение таких классификаторов повсеместно.

- Ошибки в выборке. По причине того, что частично выборка состоит из случайных примеров, весьма вероятно, что существует небольшое число пар, которые ошибочно промаркированы отрицательными. Это может понизить качество обученной модели. Помимо этого могут возникать ошибки второго рода. Например, если внешний сервис, предоставляющий переводы для слов и фраз, допустил ошибку, то в множество положительных примеров попадет неправильный перевод для слова.

В связи с описанными выше недостатками автором были проведены исследования возможности генерации обучающей выборки в автоматическом или полуавтоматическом режиме. Результаты этих исследований детально описаны в следующем разделе.

Алгоритм автоматизированной генерации обучающей выборки

Для устранения недостатков, описанных в предыдущем разделе, был разработан метод автоматической генерации обучающей выборки. Целью, как и ранее, является определение множества пар ключевых слов с соответствующими оценками $((w_{i1}, w_{i2}), y_i)_{i=1, \dots, N}$, где $w_{ij} \in W$ — слова из множества всех ключевых слов информационно-аналитической системы; число $y_i \in \{0, 1\}$ указывает на то, близки ли семантически соответствующие слова (позитивные или положительные примеры) или нет (негативные или отрицательные примеры). Суть данного алгоритма заключается в использовании графовой структуры данных для получения обучающих примеров. Для построения необходимых графов используются

понятия графа ключевых слов и полного контекстного графа, подробно описанные в работе [9].

Рассмотрим алгоритм генерации обучающей выборки на упрощенном примере. Допустим, коллекция наборов ключевых слов T состоит из приведенных далее наборов:

$$\begin{aligned} & [X, Y, W, A, C] \\ & [A, B, V, W] \\ & [A, C, Z, W] \\ & [C, Y, Z, W]. \end{aligned}$$

Здесь каждая заглавная буква подразумевает одно ключевое слово. Различным буквам соответствуют различные ключевые слова. По этим данным имеется возможность построить некоторый граф, вершинами которого служат ключевые слова, а ребра отражают некоторую степень близости пары ключевых слов. Примерами таких графов являются граф ключевых слов G_{kw} и усеченный контекстный граф G_{cont} , описание которых приведено в работе [9]. Допустим, был выбран некоторый алгоритм построения графа. Изображение этого графа приведено на рис. 1.

Для простоты изложения веса ребер на приведенной иллюстрации опущены. Отметим также, что структура графа выбрана исключительно для наглядности примера и предполагается, что существование ребра между двумя вершинами говорит о некоторой "похожести" соответствующих ключевых слов. Следующим шагом алгоритма является искусственная модификация исходной коллекции T . Для этого проводятся следующие операции:

- фиксируется дискретное распределение с неотрицательными значениями P ;

- для каждого вхождения каждого слова коллекции T из распределения P выбирается случайное число r ;

- если $r > 0$, то рассматриваемое вхождение слова w заменяется на w^{*r} .

В программной реализации алгоритма генерации обучающей выборки для модификации ключевого слова к нему в конец добавляется специальный символ, который не используется в коллекции, с последующим добавлением номера r , выбранным случайно. Данный способ замены гарантирует уникальность нового ключевого слова среди множества ключевых слов W изначальной коллекции T .

Рассматривая для упрощения в качестве дискретного распределения равномерное распределение из

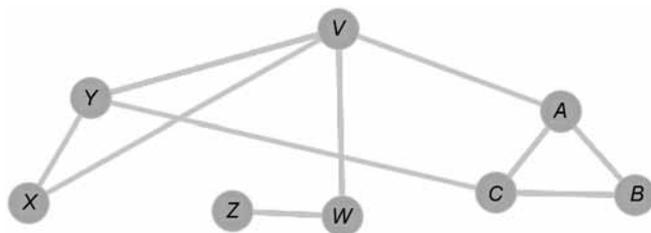


Рис. 1. Граф, построенный по упрощенной коллекции ключевых слов

двух значений, исходная коллекция может быть модифицирована, например, следующим образом:

[X, Y, W, A1, C]
 [A, B1, V, W]
 [A1, C1, Z, W]
 [C, Y, Z, W].

Обозначим модифицированную коллекцию T^m . Используя ту же процедуру, построим новый граф по модифицированной коллекции. Допустим, что в итоге был сконструирован граф, вид которого изображен на рис. 2.

Можно заметить, что в ходе работы алгоритма были модифицированы вершины A , B и C . После такого перестроения в графе появились соответствующие им вершины $A1$, $B1$ и $C1$. Однако эти пары не обязаны соединяться ребром в этом графе. Вследствие модификации данных новый граф мог утратить некоторые старые связи (например, ребро (A, V)). Новые вершины оказались связаны с другими вершинами графа.

В процессе модификации семантическому изменению слова не подверглись, поскольку к ним были приписаны служебные символы в конец. Другими словами, пара слов $(A, A1)$ является, по сути, одним и тем же словом, и, следовательно, может быть положительным обучающим примером для выборки, потому что для этих слов существуют две вершины в графе. Таким образом, для рассмотренного примера определены положительные обучающие примеры $(A, A1)$, $(B, B1)$, $(C, C1)$.

Отметим, что в рамках данной статьи в качестве процедуры построения графа был выбран описанный в работе [9] процесс построения графа ключевых слов G_{kw} . Версия этого графа, построенная по модифицированной выборке, далее обозначается как G_{kw}^m .

Задачей машинного обучения в данном случае является восстановление факта идентичности различных версий одного слова путем выявления различных графовых связей между парами обучающих примеров. Под графовыми связями здесь и далее подразумеваются различные меры и характеристики, которые можно вычислить для пары вершин одного графа. Такими характеристиками являются длины кратчайших путей, число общих соседей между вершинами, факт принадлежности одной компоненте связности и пр. Далее используется предположение

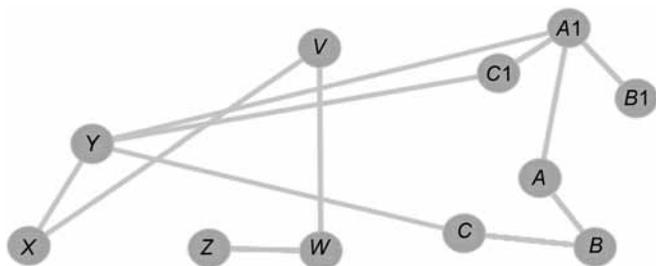


Рис. 2. Граф, построенный по модифицированной упрощенной коллекции ключевых слов

о том, что найденные закономерности, влияющие на значение уровня семантической близости в графе G_{kw}^m , также присутствуют в графе G_{kw} . Это предположение имеет место по той причине, что графы G_{kw}^m и G_{kw} строятся с помощью одного и того же алгоритма.

Важную роль в разработанном алгоритме генерации выборки играет распределение, из которого берутся индексы модифицированных объектов. Использование бернуллиевского распределения с вероятностью успеха $p = 0,5$ имеет существенный недостаток. Этот недостаток состоит в том, что ключевые слова w и w_1 будут встречаться примерно одинаковое число раз в коллекции T^m . Таким образом, обученная модель будет предполагать, что пара хороших синонимов в исходной коллекции T^m также обладает этим свойством, что неверно. Проиллюстрировать это можно следующим примером. Слова "жалованье" и "зарплата" являются синонимами, однако очевидно, что слово "жалованье" является устаревшим и в наше время употребляется значительно реже.

С учетом приведенных замечаний для исследований было выбрано геометрическое распределение. Свойства этого распределения помогают преодолеть описанный ранее недостаток. Кроме того, за счет бесконечного носителя это распределение позволяет генерировать множество различных вариаций для одного слова (число различных вариаций для одного слова ограничено лишь числом употреблений исходного слова в коллекции T). Этот факт является важным достоинством этого распределения по двум причинам. Во-первых, такое положение дел более приближено к реальности, потому что ключевое слово может иметь более, чем одно близкое по смыслу слово. Во-вторых, это позволяет существенно обогатить выборку, поскольку любую пару модификаций исходного слова можно считать положительным примером для обучения. На рис. 3 показана гистограмма индексов при генерации модифицированных версий некоторого ключевого слова.

После того как положительные примеры обучающей выборки собраны, необходимо создать также

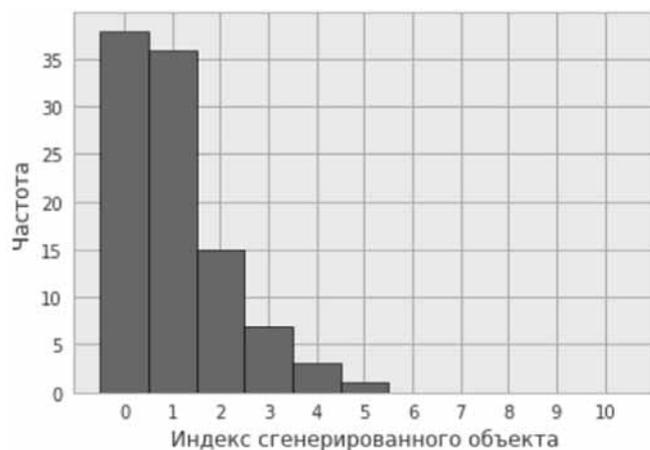


Рис. 3. Зависимость числа объектов от сгенерированного индекса

и отрицательные примеры. Для отрицательных примеров используется следующий алгоритм:

1) построение графа ключевых слов \hat{G} по модифицированному множеству ключевых слов;

2) для каждого слова w , попавшего в множество пар положительных примеров, рассматриваются случайные k соседей v_1, v_2, \dots, v_k на расстоянии 2 в графе \hat{G} ;

3) все пары $(w, v_i)_{i=1}^k$ принимаются за отрицательные примеры.

Мотивация для использования этого алгоритма заключается в следующем. Для обучения модели отрицательные примеры должны быть содержательными. Если рассмотреть абсолютно случайные пары ключевых слов, то большинство полученных примеров не будут представлять особой значимости для модели. Причина в том, что большинство статистических характеристик в этом случае будут иметь нулевое или близкое к нему значение. Между случайными вершинами в графе может не быть путей, они могут не иметь общих соседей, могут быть расположены на слишком большом удалении друг от друга и т. д. Такие случайные пары не дают никакой информации для модели для более сложных случаев, когда слова находятся достаточно близко друг к другу в графе, но при этом не являются семантически близкими. Однако именно такие случаи и представляют интерес в рамках решаемой задачи.

Вместе с тем, если для заданного ключевого слова в качестве отрицательных примеров брать непосредственных соседей в графе (т. е. максимально близкие вершины), то слишком часто будут появляться пары, которые в действительности близки по смыслу. Это происходит потому, что среди слов одного набора могут встречаться похожие слова. Вследствие этого, среди отрицательных примеров будет большое число примеров, которые ошибочно были промаркированы отрицательными, что не позволит качественно обучить модель.

Поэтому соседи вершины на расстоянии 2 представляются наиболее подходящим компромиссом: среди них не так часто встречаются семантически близкие пары, однако при этом они несут в себе содержательный сигнал, на котором можно эффективно обучаться. С учетом изложенных выше соображений алгоритм автоматизированной генерации обучающей выборки выглядит следующим образом.

1. Входные параметры: N, p, k, D .

2. Для каждого набора ключевых слов $t \in T$ системы D :

— для каждого *вхождения* ключевого слова w :

• если число вхождений в коллекцию превосходит пороговое значение N :

◇ из геометрического распределения с параметром p выбирается число;

◇ вхождение ключевого слова w заменяется w_p ;

◇ слово w_p заносится в список модификаций для слова w ;

• иначе

◇ вхождение ключевого слова w заменяется w_0 ;

— модифицированная версия набора ключевых слов добавляется в множество D^m .

3. Для каждого уникального ключевого слова $w \in W$:

— каждую пару модификаций (w_i, w_j) ключевого слова w добавить в обучающее множество X_{train} в качестве положительного примера.

4. Для модифицированной версии коллекции наборов ключевых слов D^m провести построение графа ключевых слов G_{kw}^m .

5. Для каждой вершины w_i из множества модифицированных вершин графа G_{kw}^m :

— выделить множество $G_{kw,2}^m = neighbors(G_{kw}^m, w_i, 2)$ соседей вершины w_i в графе G_{kw}^m на расстоянии 2;

— из множества $G_{kw,2}^m(w_i)$ выбрать случайно k вершин v_1, \dots, v_k и для каждой вершины добавить пару (w_i, v_j) в X_{train} в качестве отрицательного примера.

6. Вернуть обучающее множество X_{train} , граф G_{kw}^m .

Значения параметров N, p, k были определены как 10, 0,9, 5, соответственно. Необходимость выбора параметра N обусловлена тем наблюдением, что если слово встречается в коллекции D слишком редко, то всего его модификации будут встречаться в коллекции D^m еще реже.

Для этих слов в графах будет мало связей. Поэтому те слова w , которые встретились в исходной коллекции меньше, чем N раз, игнорируются алгоритмом, а точнее, всегда заменяются на w_0 .

Вхождением ключевого слова в описанном выше алгоритме является появление ключевого слова в одном конкретном наборе ключевых слов. Поскольку одно слово может присутствовать во многих наборах, то число вхождений равно числу наборов, содержащих данное ключевое слово.

Следующее далее утверждение дает оценку вычислительной сложности алгоритма генерации обучающей выборки.

Утверждение 1. Расчет искусственной обучающей выборки имеет сложность $O(f^2k + w^2k^2)$, где f — максимальная частотность слова в коллекции; k — число уникальных слов в коллекции; w — максимальный размер набора в коллекции ключевых слов.

Лемма 1. Построение положительных примеров искусственной обучающей выборки имеет сложность $O(f^2k)$, где f — максимальная частотность слова в коллекции; k — число уникальных слов в коллекции.

Доказательство. Число всех ключевых слов системы с повторениями не превышает значение fk . Необходимо пройти один раз по всем ключевым словам коллекции, чтобы сгенерировать для них модифицированную версию — это требует $O(fk)$ операций. При этом предполагается, что выбор случайного числа из выбранного распределения занимает константное время. Следующим шагом необходимо из модифицированных версий каждого слова получить всевозможные пары. Для каждого из k слов может быть получено не более, чем f модификаций слова, поскольку каждая модификация была получена из некоторого исходного слова, которое, в свою очередь, было использовано не более, чем f раз в коллекции. Для генерации всех пар для данного исходного слова

необходимо $O(f^2)$ времени, а для всех уникальных слов — $O(f^2k)$.

Лемма 2. Построение отрицательных примеров искусственной обучающей выборки имеет сложность $O(w^2k^2)$, k — число уникальных слов в коллекции; w — максимальный размер набора в коллекции ключевых слов.

Доказательство. Построение графа ключевых слов занимает не более $O(k^2)$ времени. Далее необходимо рассчитать соседей для каждой вершины на расстоянии 2. Для этого необходимо возвести в квадрат матрицу смежности графа. При хранении графа в виде разреженной матрицы операция умножения матрицы размера $(k \times k)$ на себя требует $O(nnz \cdot k)$ операций, где nnz — число ненулевых элементов. Поскольку в графе ключевых слов ребрами являются те пары ключевых слов, которые входят в один набор, то число ребер в графе, как и число ненулевых элементов в матрице смежности, не превышает $O(w^2 \cdot k)$, а сложность поиска соседей второго порядка будет равна $O(w^2 \cdot k^2)$. Предполагая, что операция взятия случайного числа из заданного распределения занимает константное время, получаем, что генерация отрицательных примеров для искусственной обучающей выборки требует $O(w^2 \cdot k^2)$ операций.

Из двух доказанных выше лемм следует утверждение 1. После того как обучающие выборки созданы, необходимо для обучающих объектов посчитать различные графовые характеристики, которые будут описывать данный объект. Вместе с целевой переменной эти характеристики необходимы для обучения классифицирующей модели.

Разработанный подход к определению смысловой близости с помощью машинного обучения и автоматизированной обучающей выборки имеет следующую особенность. Обученная на графе G_{kw}^m модель впоследствии используется для предсказания семантически близких пар уже на графе G_{kw} . Это накладывает некоторые ограничения на признаки, которыми описываются объекты машинного обучения. В следующем разделе приведено описание этих признаков и объяснена отмеченная особенность.

Признаковое описание модели машинного обучения

Под признаковым описанием пары ключевых слов понимается определение набора функций, вычисляющих некоторые числовые характеристики для рассматриваемых слов. Такие функции (факторы) могут использовать оба слова в явном виде (расстояние Левенштейна, число общих слов и т. д.) или только одно из слов (частотность данного слова в корпусе, число символов в слове и т. д.), что порождает пару значений признака для левого и правого слов из пары.

Результатом применения программных реализаций функций к паре ключевых слов является фиксированный по длине массив чисел. Массивы чисел, посчитанные для всех пар ключевых слов обучающей выборки, образуют так называемую матрицу объектов

признаков. Строки этой матрицы представляют собой признаковое описание обучающих объектов выборки, а столбцы соответствуют признакам.

Признаком может являться любая функция, способная вычислить значение для поданного на вход объекта. Однако следует понимать, что не каждая функция может быть полезна в процессе обучения. Формирование списка признаков, наилучшим образом характеризующих объекты, является важным шагом при решении задачи машинного обучения с использованием модели градиентного бустинга. Массив признаков представляет собой то, как "видит" объект модель машинного обучения. В связи с этим признаки должны в полной мере описывать объект. В рамках поставленной задачи это означает, что для пары ключевых слов полезными могут быть признаки, описывающие различные синтаксические, морфологические, семантические, статистические и графовые характеристики пары рассматриваемых слов.

Таким образом, в ходе работы алгоритма для всех рассматриваемых пар ключевых слов происходит вычисление всего набора факторов. В результате появляется матрица объект—признак. Строка этой матрицы является описанием конкретной пары ключевых слов, а столбец определяет значения некоторого признака для всех пар. Для обучающей выборки кроме матрицы объект—признак также имеется вектор-столбец истинных значений. Это позволяет обучать алгоритмы машинного обучения, после чего применять обученные модели для предсказания на новых данных (для которых также можно получить признаковое описание). Отметим тот факт, что описанные в предыдущем разделе методы формирования обучающей выборки вносят существенные ограничения на разработку признаков. В рамках данной работы выделяются два принципиально различных класса признаков: графовые и языковые.

Графовыми факторами являются те, которые невозможно получить исключительно по графам. Графовые признаки ни в каком виде не учитывают содержимое вершин графа, т. е. не учитывают особенности ключевого слова с точки зрения естественного языка. Вместо этого, такие признаки используют только связи данной вершины с другими вершинами.

Языковые признаки, напротив, используют знания из области обработки естественного языка — морфологические, синтаксические и семантические. Кроме того, к классу языковых были отнесены признаки, вычисленные с помощью внешних по отношению к рассматриваемой информационно-аналитической системе наборов данных.

Необходимость выбранного разделения признаков на классы обусловлена следующим обстоятельством. В связи с особенностями процесса формирования автоматизированной обучающей выборки для моделей, обученных на этом типе выборки не представляется возможным использование языковых факторов. Причиной этому является принцип построения автоматизированной обучающей выборки. Для генерации нового положительного примера к рассматриваемому слову приписывается служебный, семантически

незначаший суффикс. Это означает, что языковые различия между словами в такой паре являются искусственными и не отражают действительную природу вещей. Использование эвристической обучающей выборки не накладывает ограничений на выбор признаков, однако имеет недостатки, описанные ранее.

После вычисления значений всех признаков начинается процесс обучения модели. Для его запуска необходимо подать матрицу объектов-признаков и обучающую выборку (эвристическую или автоматизированную) на вход модели обучения. С учетом изложенных выше соображений, для решения поставленной задачи необходимо определить множество факторов, которое позволит достичь наилучшего показателя качества модели машинного обучения. Для автоматизированной выборки выделяются графовые факторы, а для эвристической — графовые и языковые факторы. Особый интерес представляет автоматизированная выборка, поскольку ее использование позволяет полностью автоматизировать весь процесс решения задачи семантической близости пары ключевых слов. Полный список используемых факторов представлен в приложении.

Для экспериментов была выбрана зарекомендовавшая себя в индустрии модель машинного обучения XGBoost [10].

Тестовые испытания

Далее представлены результаты тестирования программных реализаций описанных алгоритмов определения близости с использованием методов машинного обучения. Тестирование программных реализаций алгоритмов проводилось на следующих наборах данных.

- **Aminer.** Открытая коллекция наукометрических данных, созданная авторами работы [8]. Коллекция располагает англоязычной информацией о научных публикациях, авторах, конференциях и ключевых словах к научным статьям. Для тестирования была выбрана коллекция ключевых слов из категории обработки естественного языка (*natural language processing*).

- **KeywordsRu.** Коллекция наборов русскоязычных ключевых слов, собранных из сети Интернет. Используемая выборка была использована ранее в работе [9].

- **KeywordsEn.** Коллекция наборов англоязычных ключевых слов, собранных из сети Интернет. Ключевые слова для данной выборки были собраны аналогично процедуре сбора русскоязычного набора данных KeywordsRu.

Валидация качества работы программных реализаций алгоритмов была проведена с помощью введенной ранее эвристической обучающей выборки. Модели на вход подается одна из рассматриваемых коллекций ключевых слов. Программная реализация строит все необходимые графы ключевых слов, определяет на их основе автоматизированную обучающую выборку и вычисляет факторы для каждого

элемента выборки. Далее запускается процесс обучения модели. После чего происходит процесс применения обученной модели к объектам эвристической обучающей выборки. Последним этапом является вычисление значения метрик, характеризующих качество работы программной реализации.

Модель, рассматриваемая в рамках данной работы, получила название WordMLSim, в то время как модель, введенную в работе [9], далее будем называть WordContSim. Кроме того, для рассматриваемых коллекций ключевых слов используются известные модели вычисления семантической близости. В рамках этого эксперимента были выбраны уже представленная ранее модель Word2Vec, а также модель Node2Vec. В то время как Word2Vec использует знания о языке для решения задачи семантической близости, модель Node2Vec определяет близость между вершинами графа. Для этого определяют векторное представление для вершин графа. Меньшее расстояние между векторами влечет более выраженную структурную связь между соответствующими вершинами. Следует также отметить, что модель Node2Vec переносит идеи, заложенные в модель Word2Vec, с текстовых данных на графовые.

Выбор моделей Word2Vec и Node2Vec для сравнения с моделью, представленной в данной работе, обусловлен следующими соображениями. В первую очередь было необходимо продемонстрировать высокий уровень определения семантической близости между словами естественного языка. Для этой цели авторская модель сравнивается с зарекомендовавшим себя в индустрии решением — моделью Word2Vec. Кроме этого, поскольку в ходе работы используются графовые методы определения близости, является важным сравнить реализацию модели с известными решениями вычисления близости между вершинами графа. В качестве такого решения выступает модель Node2Vec.

В дополнение к этому представляет интерес следующее рассуждение. Разработанные признаки являются функциями от пары вершин графа, и, следовательно, каждый признак может представлять некоторую меру близости пары ключевых слов. Подобно этому, модель WordContSim, с одной стороны, является признаком для модели WordMLSim, а с другой стороны, оказывается самодостаточным подходом к определению смысловой близости, как было показано в работе [9]. В связи с этим важно проверить, что модель, обученная на всем множестве факторов, качественно превосходит любой из признаков, входящих в ее состав. Этот факт докажет целесообразность применения машинного обучения в рамках поставленной задачи.

В таблице представлены результаты тестирования моделей семантической близости. Для определения уровня качества использовались метрики NDCG¹ и AUC².

¹ https://en.wikipedia.org/wiki/Discounted_cumulative_gain

² https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Результаты тестирования моделей семантической близости

Модель	Коллекция					
	Aminer		KeywordsRu		KeywordsEn	
	AUC	NDCG	AUC	NDCG	AUC	NDCG
WordMLSim	0,6306	0,1666	0,7718	0,1073	0,8825	0,3336
Word2Vec	0,9475	0,2326	0,8822	0,1402	0,8506	0,3431
Node2Vec	0,6015	0,1658	0,7511	0,1015	0,8254	0,3109
Лучший признак	0,6275	0,1667	0,7549	0,1064	0,8606	0,3286
WordContSim	0,5979	0,1668	0,7279	0,1066	0,8103	0,3272

Наилучший результат, как и ожидалось, достигается применением модели Word2Vec, поскольку только эта модель опирается на заложенные в нее знания о языке. Остальные протестированные модели используют только информацию о коллекции, представленную в виде графа. Таким образом, модель Word2Vec получает неоспоримое преимущество в рамках данного эксперимента. Отметим, что результаты тестовых испытаний на коллекции KeywordsEn демонстрируют конкурентоспособность разработанной модели WordMLSim, даже без наличия каких-либо знаний о языке. По метрике AUC разработанная автором модель превзошла известный в индустрии аналог.

Среди моделей, использующих исключительно знания о тестируемой коллекции, модель WordMLSim превосходит модель Node2Vec. Данный факт свидетельствует о высоком уровне качества разработанной модели определения семантической близости. Кроме этого, WordMLSim показывает лучшие результаты в сравнении как с разработанной в работе [9] моделью WordContSim, так и с отдельными признаками, использованными для обучения. Данное обстоятельство еще раз подтверждает целесообразность использования методов машинного обучения для определения уровня семантической близости.

Заключение

Описаны методы автоматизированного формирования обучающей выборки на основе графового представления данных. На основе собранных примеров были представлены модели машинного обучения, решающие задачу определения семантической близости пары слов. Авторский алгоритм позволяет избежать трудоемкого процесса сбора данных для обучения моделей. Модели, обученные на данных автоматизированной выборки, показывают высокий уровень качества. Процесс подготовки модели полностью автоматизирован. Разработанная модель WordMLSim продемонстрировала конкурентоспособный уровень качества в сравнении с моделями, использующими настоящую обучающую выборку, и моделями, обученными на огромных наборах данных. Используемые методы машинного обучения

помогают избавиться от необходимости разработки эвристических моделей. Модели, построенные по автоматизированной обучающей выборке, позволяют решать не только задачу определения семантической близости, но и произвольную задачу классификации ребер в графе.

Недостатком работы может являться выбор отрицательных примеров для выборки, поскольку опирается на процедуру выбора некоторой случайной вершины в окрестности рассматриваемой вершины. Такое положение дел может приводить к присутствию ложноотрицательных примеров в выборке и последующему ухудшению качества модели. Поэтому дальнейшим направлением в изучении данной области может служить разработка алгоритмов автоматического определения отрицательных обучающих примеров.

Автор выражает благодарность д-ру физ.-мат. наук, проф. В. А. Васенину и канд. физ.-мат. наук, вед. науч. сотр. С. А. Афонину за внимание к работе и помощь в подготовке статьи.

Список литературы

1. Mikolov T., Sutskever I., Chen K. et al. Distributed Representations of Words and Phrases and Their Compositionality // Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13). Vol. 2. — Lake Tahoe, Nevada, Curran Associates Inc., 2013. — P. 3111–3119. URL: <http://dl.acm.org/citation.cfm?id=2999792.2999959>
2. Pennington J., Socher R., Glove C. D. Global vectors for word representation Manning // In EMNLP. — 2014. — P. 1532–1543.
3. Wu L., Fisch A., Chopra S. StarSpace: Embed All The Things! URL: <http://arxiv.org/abs/1709.03856>
4. Gidaris S., Singh P., Komodakis N. Unsupervised representation learning by predicting image rotations // In International Conference on Learning Representations (ICLR), 2018. URL: <http://arxiv.org/abs/1803.07728>
5. Doersch C., Gupta A., Efros A. A. Unsupervised visual representation learning by context prediction // In International Conference on Computer Vision (ICCV), 2015. — P. 1422–1430. URL: <https://arxiv.org/pdf/1505.05192.pdf>
6. Grover A., Leskovec J. Node2vec: Scalable Feature Learning for Networks // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16). — San Francisco, California, USA: Association for Computing Machinery, 2016. — P. 855–864.
7. Kipf T., Welling M. Semi-Supervised Classification with Graph Convolutional Networks // Proceedings of the 5th International Conference on Learning Representations, 2017. URL: <https://arxiv.org/abs/1609.02907>
8. Tang J. AMiner: Mining Deep Knowledge from Big Scholar Data // Proceedings of the 25th International Conference Companion on World Wide Web. — Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016. — P. 373.
9. Лунев К. В. Графовые методы определения семантической близости пары ключевых слов и их применения к задаче кластеризации ключевых слов // Программная инженерия. — 2018. — Т. 9, № 6. — С. 262–271.
10. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16). — San Francisco, California, USA: ACM, 2016. — P. 785–794.

Algorithm for Automated Generation of a Training Sample for Solving the Problem of Determining Semantic Similarity between a Pair of Keywords using Machine Learning Methods

K. V. Lunev, kirilllunev@gmail.com, Moscow State University, Moscow, 119192, Russian Federation

Corresponding author:

Lunev Kirill V., Postgraduate Student, Moscow State University, Moscow, 119192, Russian Federation
E-mail: kirilllunev@gmail.com

Received on May 10, 2021

Accepted on May 31, 2021

Currently, machine learning is an effective approach to solving many problems of information-analytical systems. To use such approaches, a training set of examples is required. Collecting a training dataset is usually a time-consuming process. Its implementation requires the participation of several experts in the subject area for which the training set is collected. Moreover, for some tasks, including the task of determining the semantic similarity of keyword pairs, it is difficult even to correctly draw up instructions for experts to adequately evaluate the test examples. The reason for such difficulties is that semantic similarity is a subjective value and strongly depends on the scope, context, person, and task. The article presents the results of research on the search for models, algorithms and software tools for the automated formation of objects of the training sample in the problem of determining the semantic similarity of a pair of words. In addition, models built on an automated training sample allow us to solve not only the problem of determining semantic similarity, but also an arbitrary problem of classifying edges of a graph. The methods used in this paper are based on graph theory algorithms.

Keywords: semantic similarity, natural language processing, graph algorithms, graph theory, machine learning

For citation:

Lunev K. V. Algorithm for Automated Generation of a Training Sample for Solving the Problem of Determining Semantic Similarity between a Pair of Keywords using Machine Learning Methods, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 6, pp. 383—294.

DOI: 10.17587/prin.12.383-294

References

1. Mikolov T., Sutskever I., Chen K. et al. Distributed Representations of Words and Phrases and Their Compositionality, *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, Lake Tahoe, Nevada, Curran Associates Inc., 2013, pp. 3111—3119, available at: <http://dl.acm.org/citation.cfm?id=2999792.2999959>
2. 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—1543.
3. Wu L., Fisch A., Chopra S. StarSpace: Embed All The Things!, available at: <http://arxiv.org/abs/1709.03856>, 2017
4. Gidaris S., Singh P., Komodakis N. Unsupervised representation learning by predicting image rotations. *Proceedings of the 2018 International Conference on Learning Representations (ICLR)*, 2018, available at: <http://arxiv.org/abs/1803.07728>,
5. Doersch C., Gupta A., Efros A. A. Unsupervised visual representation learning by context prediction, *Proceedings of the 2015 International Conference on Computer Vision (ICCV)*, 2015, pp. 1422-1430, available at: <https://arxiv.org/pdf/1505.05192.pdf>
6. Grover A., Leskovec J. Node2vec: Scalable Feature Learning for Networks, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, Association for Computing Machinery, 2016, pp. 855—864.
7. Kipf T., Welling M. Semi-Supervised Classification with Graph Convolutional Networks, *Proceedings of the 5th International Conference on Learning Representations*, 2017, available at: <https://arxiv.org/abs/1609.02907>
8. Tang J. AMiner: Mining Deep Knowledge from Big Scholar Data, *Proceedings of the 25th International Conference Companion on World Wide Web*, Montréal, Québec, Canada, International World Wide Web Conferences Steering Committee, 2016, pp. 373.
9. Lunev K. V. Graph methods for determining the semantic similarity of a pair of keywords and their application to the problem of keywords clustering, *Programmnyaya Ingeneriya*, 2018, vol. 9, no. 6, pp. 262—271 (in Russian).
10. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2016, pp. 785—794.

Список используемых факторов при обучении моделей

Признак	Описание	Тип
Расстояние в графе ключевых слов	—	Парный/графовый
Расстояние в усеченном графе ключевых слов	—	Парный/графовый
Контекстная близость	Значение контекстной близости из работы [9]	Парный/графовый
Ранг контекстной близости	На каком месте данное правое слово находится среди всех слов для данного левого слова (при сортировке по уровню контекстной близости)	Парный/графовый
Число вершин в кластере	—	Одиночный/графовый
PageRank вершины графа	—	Одиночный/графовый
EdgeRank ребра графа	—	Парный/графовый
Вес ребра в графе ключевых слов	—	Парный/графовый
Частотность левого слова	—	Одиночный/графовый
Частотность правого слова	—	Одиночный/графовый
Совместная частотность в корпусе	—	Парный/графовый
<i>Pointwise mutual information</i>	—	Парный/графовый
Мера Жаккара	Считается по вершинам-соседям	Парный/графовый
Число простых путей в графах между вершинами	—	Парный/графовый
<i>Betweenness centrality</i>	—	Одиночный/графовый
<i>Closeness centrality</i>	—	Одиночный/графовый
<i>Eigen vector centrality</i>	—	Одиночный/графовый
Значение потока между вершинами	—	Парный/графовый
Число соседей	На расстоянии 1 и 2 (два фактора)	Одиночный/графовый
Число общих соседей	На расстоянии 1 и 2 (два фактора)	Парный/графовый
Расстояние Левенштейна	—	Парный/языковой
Часть речи слова	Категориальный признак	Одиночный/языковой
Число слов	—	Одиночный/языковой
Число букв	—	Одиночный/языковой
Число общих слов	—	Парный/языковой
Пословная мера Жаккара	—	Парный/языковой
<i>N</i> -граммная мера Жаккара	Используется биграммное и триграммное представление слов	Парный/языковой
Пара слов на разных языках	Бинарный признак	Парный/языковой
Одно слово является аббревиатурой/сокращением для другого	Бинарный признак	Парный/языковой
Одно слово является формой другого	Бинарный признак	Парный/языковой
Одно слово является транслитерацией другого	Бинарный признак, использован наивный алгоритм транслитерации	Парный/языковой
Расстояние по дереву WordNet	—	Парный/языковой
Глубина слова в дереве WordNet	—	Одиночный/языковой
Косинусное расстояние между word2vec-представлениями слов	—	Парный/языковой
Средняя частотность слов в корпусе	В качестве слов рассматриваются не ключевые слова, а отдельные слова естественного языка	Одиночный/языковой
Косинусное расстояние пословных TFIDF-представлений	В качестве слов рассматриваются не ключевые слова, а отдельные слова естественного языка	Парный/языковой
Косинусное расстояние <i>n</i> -граммных TFIDF-представлений	Подсчитано на символьных триграммах	Парный/языковой

А. А. Титова, науч. сотр., titova_a@maltsystem.com,
В. А. Роганов, канд. физ.-мат. наук, науч. сотр.,
Г. А. Лукьянченко, канд. физ.-мат. наук, ст. науч. сотр.,
С. Г. Елизаров, канд. физ.-мат. наук, руководитель дизайн-центра,
Дизайн-центр цифровой микроэлектроники Физического факультета
МГУ им. М. В. Ломоносова

Повышение эффективности транзакций в блокчейн-сетях, использующих алгоритм Cryptonight в качестве Proof of Work на базе специализированного многоядерного процессора MALT

Предложен способ оптимизации memory-bound-алгоритма Cryptonight, используемого в качестве Proof of Work в криптовалютах. Создана реализация алгоритма для специализированного многоядерного процессора MALT [1] с учетом преимуществ его архитектуры. Описаны оптимизации, примененные для программного ускорения этой реализации алгоритма — кластеризация и локализация данных, использование специализированного сопроцессора для параллельных вычислений, предварительная загрузка данных. Основанные на тестах и теоретических расчетах результаты показывают, что энергоэффективность удалось увеличить в 5 раз по сравнению с процессором общего назначения.

Ключевые слова: memory-bound-алгоритмы, майнинг, многоядерные процессоры, специализированные процессоры, параллельные вычисления, SIMD-вычисления, Cryptonight, криптовалюты

Введение

Криптовалюта — один из видов цифровой валюты, создание и контроль которой базируются на криптографических методах шифрования. Функционирование этой системы основано на технологии блокчейна. Для обеспечения работы системы ее участники-добровольцы занимаются майнингом — создают новые блоки, проводя математические и криптографические вычисления на своем компьютере, и получают за них вознаграждение в виде единиц валюты. Данный алгоритм называется Proof of Work — чтобы получить криптовалюту, участник системы должен отправить "подтверждение работой", доказать, что потратил ресурсы на вычисления.

На данный момент самым эффективным методом майнинга криптовалют, использующих алгоритм Proof of Work, является использование ASIC (Application-Specific Integrated Circuit) — интегральных схем, специализированных для решения конкретной задачи. Особенности решения задач для заработка криптовалют позволяют применить максимальное распараллеливание вычислений практически для всех используемых алгоритмов. ASIC-устройства специализированы под конкретную криптовалюту и способны выполнять вычисления только для одного заданного алгоритма, но делают это быстрее

и более энергоэффективно, чем любые другие виды устройств для майнинга [2].

Вследствие распространения ASIC-устройств с 2012—2013 гг. майнинг распространенных криптовалют на неспециализированной технике стал невыгодным. Требования к оборудованию выросли, и эффективно создавать новые блоки могло лишь малое число высокопроизводительных устройств, сосредоточенных в руках небольшого числа людей, занимающихся майнингом — майнеров. Единственной альтернативной возможностью для тех, кто не входил в их число, стали майнинговые пулы — объединения майнеров, в которых вычисления разделяются на всех участников, а вознаграждение распределяется пропорционально вложенному усилию.

Система криптовалюты может корректно работать только до тех пор, пока большинство ее пользователей честны, однако в силу возросшей сложности вычисления, большая часть вычислительных мощностей оказалась сконцентрирована у очень небольшого числа майнеров и майнинговых пулов. Это привело к централизации майнинга и потенциально грозило нарушить защиту блокчейна от подделок и повторной траты одних и тех же средств — консенсус большинства может быть достигнут, если несколько майнеров или пулов объединятся, получив контроль над всей сетью [3]. В этом случае новые

блоки будут генерироваться уже не по принципам криптовалюты, а по их усмотрению [4], но при этом принципы блокчейна нарушены не будут, поэтому технически предотвратить такой исход событий, не отказавшись от принципа Proof of Work, невозможно.

Одним из решений этой проблемы стало создание новых алгоритмов, делающих невыгодным использование специализированного оборудования, уравнивающим его пользователей и пользователей обычных персональных компьютеров. Это алгоритмы, зависящие от памяти (*memory-bound*). Таких алгоритмов довольно много, но их объединяет общая черта: в процессе вычислений используется массив памяти размером в несколько мегабайт, случайные чтения и записи в который очень замедляют вычисления для ASIC-устройств, не обладающих достаточно большим кешем.

Таким образом, возможность увеличения скорости расчетов при майнинге криптовалют, основанных на *memory-bound*-алгоритмах, на специализированных программируемых процессорах является перспективным направлением исследования. Как правило, для повышения скорости и энергоэффективности таких расчетов вместо универсальных процессоров разрабатывают непрограммируемые специализированные ASIC-устройства с применением аппаратных оптимизаций. Эти методы исключительно эффективны для простых криптовалют, в которых необходимо многократно и независимо повторить одно стандартное криптопреобразование, но не всегда подходят для криптовалют со сложным, динамически изменяемым алгоритмом расчета или алгоритмом расчета, требующим нерегулярной работы с памятью достаточно большого объема.

Важным критерием, определяющим возможность эффективного переноса криптоалгоритма на ASIC, является наличие в конкретной реализации майнера полного параллелизма, т. е. возможности параллельно и полностью независимо выполнять множество вычислительных потоков, так как это позволяет встроить в ASIC тысячи идентичных сверхпроизводительных вычислителей и раздать им части задачи. Кажется, что любой алгоритм майнинга удовлетворяет этому критерию, так как всегда сводится к задаче парольного опробования, суть которой в многократном и независимом повторении одного и того же преобразования над разными наборами входных данных. Это верно, но только в случае, когда все используемые при работе каждого сверхпроизводительного вычислителя переменные локальны.

Увеличивая требования к общему объему памяти, требуемому специализированному прибору для майнинга для вычислений, разработчики некоторых криптовалют заставляют разработчиков таких приборов уходить от использования накристалльной памяти ASIC и искать решения с внешней относительно вычислительного чипа памятью типа DDR (*Double Data Rate*). Это обычно означает переход от ASIC к GPU, обладающему большим объемом лучшей на рынке памяти типа DDR с огромной пропускной способностью в режиме последовательного чтения. Требуя перевода внешней памяти в режим

чтения/записи по произвольным адресам, разработчики криптовалют заставляют разработчиков приборов для майнинга идти дальше и переходить на CPU. Для закрепления успеха иногда требуемый объем памяти фиксируется на уровне объема L3 кеша современных CPU, совмещающего объем и скорость на произвольном доступе. Таким образом, разработчики криптовалют, вводя "узкое место" — внешнюю память и интерфейс к ней, существенно ограничивают возможности параллелизма и, соответственно, применения специализированных процессоров для майнинга. Декларируемая цель — "честная" децентрализация типа "один процессор — один голос".

Не будем анализировать эту цель, лучше рассмотрим одного из наиболее ярких представителей этого подхода — алгоритм Cryptonight, который был разработан так, чтобы к нему нельзя было применить ни внешний, ни внутренний параллелизм. Первое достигается за счет использования для вычислений массива большого размера (2 Мбайта), называемого скрэтчпадом, второе — за счет непредсказуемых обращений в память, адрес которых становится известен только перед запросом на чтение.

Алгоритм подобен остальным *memory-bound*-алгоритмам: сначала на основе исходных данных заполняется скрэтчпад, затем следует основная часть, занимающая больше всего времени — $524\,288 (2^{19})$ итераций, состоящих из случайных чтений, записей и криптопреобразований, а в конце скрэтчпад сужается до размеров хеша [5].

AES (*Advanced Encryption Standard*) используется как основной криптоалгоритм при шифровании, на его основе происходит и формирование скрэтчпада, и его дальнейшие преобразования. Помимо AES применяют и другие алгоритмы, такие как Blake, Kessac, Skein, но доля времени их исполнения ничтожно мала, поэтому их оптимизация в данной статье не рассматривается.

На настоящий момент существует не так много исследований о программной оптимизации *memory-bound*-алгоритмов. Анализ известных решений показал, что в основном научные работы по схожей тематике сосредоточены на аппаратных оптимизациях [6, 7]. Но лишь в одной статье [8] авторы анализируют несколько подобных алгоритмов, в том числе Cryptonight, указывая на трудности, с которыми придется столкнуться при попытке их оптимизации для параллельной работы, и предлагают свои методы по преодолению этих препятствий. В статье также представлены результаты экспериментальной оптимизации алгоритма Ethash, показывающие заметное улучшение по сравнению с процессором общего назначения. Таким образом, подобная работа уже существует, однако особенности каждого алгоритма не позволяют получить универсальный метод программной оптимизации, требуя более детального изучения.

На данный момент критически мало исследований вопроса о возможности эффективной реализации *memory-bound*-алгоритмов майнинга вообще и Cryptonight в частности на многоядерных специализированных процессорах, и данная статья призвана это исправить.

Характеристики используемых плат

Характеристики	СБИС MALT-C-v2	ПЛИС, моделирующая MALT-D
Частота SIMD-сопроцессоров, МГц	800	50
Частота RISC-ядер, МГц	800	100
Объем локальной памяти ядер, КБайт	8	32
Объем внешней памяти, МБайт	128	128
Латентность чтения для общей памяти, такты	50	200
Пропускная способность памяти, Гбит/с	30	55
Потребляемая мощность, Вт	5	11

В рамках работы была реализована программа, позволяющая проводить майнинг любой криптовалюты, использующей алгоритм Cryptonight v.7 в качестве базового, на специализированном многоядерном процессоре MALT, архитектура которого будет приведена далее. В статье описаны программные оптимизации, проведенные в процессе работы над кодом алгоритма. Полученные экспериментальным путем результаты доказывают, что энергоэффективность алгоритма Cryptonight v.7 можно увеличить относительно процессора общего назначения. С помощью тех же оптимизаций можно подобным образом ускорить и другие версии Cryptonight по мере их появления.

Для тестирования в работе используются СБИС (Сверхбольшая Интегральная Схема) MALT-C-v2 и ПЛИС (Программируемая Логическая Интегральная Схема) Xilinx Kintex UltraScale 115.

Инструментарий разработки

Разработка, описанная в статье, проводилась на процессорах архитектуры MALT [1]. MALT — *Multicore Architecture with Lightweight Threads* — многоядерная микропроцессорная архитектура с поддержкой легких потоков. Базовую часть архитектуры любого из процессоров MALT составляют десятки или сотни однотипных вычислительных элементов, специализированных под целевой класс задач, объединенных внутрикристалльной сетью собственной разработки.

Универсальные вычислительные ядра СБИС построены на базе процессорного ядра общего назначения — целочисленного 32-разрядного RISC-процессора. Они могут быть дополнены SIMD-сопроцессорами, используемыми для более эффективного выполнения специализированных задач целевого класса. Каждый сопроцессор содержит 8...128 однотипных процессорных элементов, имеющих общую память команд.

Все вычислительные ядра и ускорители имеют собственную локальную память данных, а также непосредственно адресуют общую внешнюю память СБИС через контроллер общих ресурсов. Кэши данных отсутствуют, поэтому достаточно велика латентность при работе с глобальной памятью.

Исполнение программы на процессоре MALT начинается с исполнения функции `main()` на управляющем ядре, которая в дальнейшем может задействовать подчиненные ядра системы. Основной задачей управляющего ядра является распределение вычислительной нагрузки по остальным ядрам.

Подчиненные ядра являются аппаратными аналогами программных тредов. Для хранения стека они используют быструю локальную память размером 8 КБайт, аналог TLS программных тредов.

В данной работе использовались два разных процессора MALT, представленные на рис. 1 (см. вторую сторону обложки): MALT-C-v2 и MALT-D. Главное их различие в том, что в MALT-D большее число RISC-ядер за счет отсутствия сопроцессоров.

Процессор MALT-D на момент написания статьи еще не выпущен в производство, поэтому для проведения данного исследования он был смоделирован с помощью ПЛИС Xilinx Kintex UltraScale 115.

В табл. 1 указаны характеристики СБИС MALT-Cv2 и реализации системы MALT-D на ПЛИС.

Использование многопоточности

Архитектура процессоров MALT подразумевает наличие одного управляющего ядра и нескольких подчиненных ядер. В данном исследовании основным ограничением числа ядер являлся объем глобальной памяти — 128 МБайта. Так как для каждого хеша необходимо 2 МБайт памяти под скрэтчпад, и еще некоторое число памяти для других целей, максимальное количество ядер, которое представляется возможным запустить параллельно на процессоре MALT-D — 62. На MALT-Cv2 подчиненных ядер всего 15, поэтому проблем с количеством памяти не возникает. Схема работы алгоритма представлена на рис. 2.

Каждый хеш вычисляется на своем ядре, вычисления никак не пересекаются. Управляющее ядро занимается распределением исходных данных по ядрам и сбором результатов.

Оптимизация работы с памятью

В качестве референтного кода для оптимизации был выбран майнер Xmrig — один из самых быстродействующих и популярных для алгоритма Cryptonight. Чтобы определить, какие области алгоритма необходимо оптимизировать для увеличения скорости исполнения и, соответственно, энергоэффективности, код был скомпилирован для MALT с небольшими изменениями и измерением времени работы. В табл. 2 представлены результаты измерения.

По сравнению с процессором общего назначения происходит довольно заметное замедление, и для

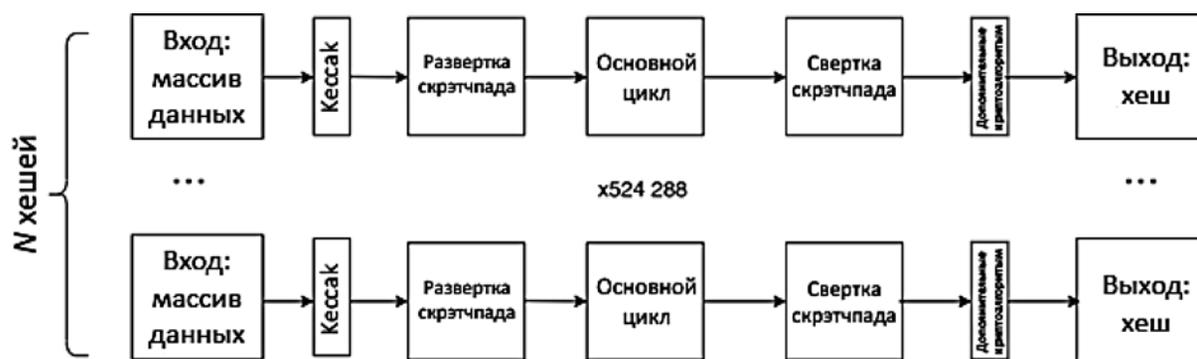


Рис. 2. Схема работы алгоритма с использованием только RISC-ядер

Таблица 2

Результаты первоначального профилирования, время исполнения, с

Процессор	
MALT-D (модель)	Intel Xeon CPU X5690 3.47 GHz
62	0,12

определения его причин было проведено более детальное профилирование, показавшее, что наиболее затратными процессами являются обращения в глобальную память.

Авторы статьи [8] при переносе нескольких memozy-bound-алгоритмов на GPU также столкнулись с этой проблемой, и результаты их исследования аналогичны. Главной причиной сильного замедления работы алгоритма являются слишком частые одиночные обращения в глобальную память, из которых состоит основной цикл. На процессоре общего назначения такого замедления не происходит, так как весь объем скрэтчпада помещается в кеш.

Таким образом, чтобы ускорить работу Cryptonight на процессорах MALT, прежде всего требовалось уменьшить число обращений к глобальной памяти. Это было достигнуто с помощью кластеризации данных и внесения в локальную память таблиц постоянных, необходимых для криптографических преобразований.

Оптимизация основного цикла

SIMD-сопроцессор проводит вычисления быстрее, чем RISC-ядра, но ограничен в количестве памяти и кода, который можно загрузить для выполнения, поэтому в данной работе он использовался лишь для некоторых вычислений. Кроме того, передача данных занимает определенное время, сравнимое с копированием в локальную память. Несмотря на эти ограничения, при переносе вычислений основного цикла на сопроцессор наблюдается увеличение скорости. Самыми затратными из вычислительных операций являются AES-преобразования, и их выполнение на RISC происходит достаточно долго, чтобы компенсировать передачу данных в сопроцессор.

Таким образом, данные проходят следующий цикл: загрузка из скрэтчпада в глобальную память в локальную память RISC-ядра, копирование в локальную память сопроцессора, AES-преобразования, копирование результата в локальную память RISC-ядра, запись в глобальную память.

После того как сопроцессор заполнен данными и вычисления запущены, он работает отдельно от RISC-ядра, которое тем временем может не простаивать, ожидая результата, а выполнять другие процессы. Это дает возможность выполнять криптопреобразования и одновременно копировать данные в сопроцессор или загружать результат в глобальную память. Но так как алгоритм Cryptonight защищен от внутреннего параллелизма, и нельзя заранее загрузить из глобальной памяти данные для следующей итерации, то сталкиваемся с проблемой простоя RISC-ядра, которому нечего делать, пока он не получит результат вычислений от сопроцессора. Чтобы избежать этого, можно подсчитывать не один, а сразу несколько хешей на одном ядре. Тогда появится возможность делать итерации по очереди, и пока блок данных одного хеша преобразуется в сопроцессоре, загрузить блок данных другого из глобальной памяти, скопировать в сопроцессор и выгрузить результат предыдущей итерации. Более того, возможности MALT позволяют загружать данные из глобальной памяти "в фоновом режиме", параллельно основному процессу, и только в схеме с несколькими хешами это преимущество может быть использовано.

Такая схема, при наличии нужного количества хешей на одном ядре, дает ускорение в несколько раз по сравнению с проведением вычислений на SIMD-сопроцессоре только для одного хеша. Но работа усложняется тем, что в основном цикле не одно, а два непредсказуемых обращения к скрэтчпаду, и их адреса не расположены рядом, поэтому невозможно загрузить данные сразу для обоих. Даже имея два хеша на ядре, избежать простоя в ожидании новых данных не получалось, поэтому было добавлено еще несколько хешей. Путем тестов и экспериментов было найдено оптимальное число хешей на одно ядро — шесть.

Модифицированный основной цикл алгоритма принимает сразу шесть хешей и вычисляет их последовательно с помощью SIMD-сопроцессора, чередуя между собой, чтобы компенсировать время, уходящее на загрузку данных из глобальной памяти.

Таблица 3

Результаты добавления SIMD-сопроцессора и схемы с вычислением нескольких хешей

Процессор	Хеш/ядро	Основной цикл, такты
MALT-D (RISC-only)	1	1 066 442 547
MALT-C (RISC + SIMD)	2	836 466 231
MALT-C (RISC + SIMD)	6	665 670 997

Использование такой схемы значительно ускорило работу алгоритма на MALT. В табл. 3 представлены результаты этой оптимизации и показано, как число вычисляемых на одном ядре хешей влияет на скорость исполнения. Как следует из табл. 3, скорость увеличилась почти в 2 раза после добавления сопроцессоров и оптимизации работы с ними.

Проведенное исследование показало, что шесть является оптимальным числом хешей и дальнейшее увеличение их числа не ведет к сокращению времени работы алгоритма.

Оптимизация работы со скрэтчпадами

Помимо основного цикла при первоначальном профилировании большую долю (30 %) занимало расширение и сжатие скрэтчпада, тогда как на процессоре общего назначения эта доля незначительна. Это объясняется тем, что обе функции представляют собой непрерывный поток обращений к глобальной памяти и AES-преобразований и отнимают большое количество времени в силу латентности памяти и сложности вычислений. Однако как и в случае с основным циклом алгоритма, для вычислений можно использовать SIMD-сопроцессор. В данном случае нет затруднений, связанных со случайными обращениями в память: в этом месте алгоритма скрэтчпад обрабатывается по порядку.

Обработка скрэтчпада проводится независимыми блоками по 128 бит, так как используется алгоритм AES-128. Эти блоки можно вычислять параллельно

друг другу. Тесты показали, что оптимальнее всего использовать четыре сопроцессора для криптопреобразований, и за время вычислений загружать из глобальной памяти следующие блоки. Увеличение числа сопроцессоров в данном случае нецелесообразно, так как ведет к увеличению простоя при копировании данных в сопроцессор и работе с глобальной памятью.

В табл. 4 приведены результаты данной оптимизации.

В данном случае удалось достигнуть улучшения в 6 раз.

На рис. 2 была представлена схема стандартного алгоритма Crypthonight, запускаемого на доступном числе подчиненных ядер. Для процессора MALT-D, в котором нет сопроцессоров, она осталась верна. Но с учетом оптимизаций, проведенных для архитектуры MALT-C, для этой версии схема будет совсем другой.

Выше уже упоминалось, что основным ограничением числа используемых ядер является объем глобальной памяти — 128 МБайт. С учетом того, что в оптимизированной версии не менее шести хешей считаются на одном ядре, максимально возможное число одновременно запущенных ядер — 10. На рис. 3 представлена новая схема работы алгоритма.

Таким образом, были получены две версии оптимизации алгоритма. Вторая подразумевает использование SIMD-сопроцессора для вычислений и считает несколько хешей на каждом ядре; первая проводит все вычисления на RISC-ядрах, используя одно для каждого хеша.

Таблица 4

Результат оптимизации работы со скрэтчпадом

Процессор	Работа со скрэтчпадом, такты
MALT-D (RISC-only)	312 452 708
MALT-C (RISC + SIMD)	52 680 723



Рис. 3. Схема работы алгоритма с использованием RISC-ядер и SIMD-сопроцессоров

Результаты

Метрикой оценки результатов работы была выбрана энергоэффективность, которая является одной из важнейших характеристик прибора для майнинга, и, в отличие от числа хешей в секунду, также учитывает затраты на подсчет хеша.

В данной работе она оценивалась по следующей формуле:

$$\frac{n_{\text{я}}v}{T(Nn_{\text{я}})},$$

где $n_{\text{я}}$ — число используемых ядер; v — частота процессора; T — число тактов, уходящих на исполнение одного хеша; N — мощность процессора.

Энергоэффективность подсчитана для двух вариантов оптимизированного алгоритма (с использованием сопроцессора и без), для процессора общего назначения INTEL CORE I7-8700 и для видеокарты XFX RADEON RX VEGA 64 (SAMSUNG). В табл. 5 представлены результаты расчета энергоэффективности.

Для СБИС MALT-C-v2 в табл. 5 представлен результат практических запусков прибора для майнинга, проводившихся на частоте 420 МГц. Видно, что на данный момент вариант с использованием SIMD-сопроцессоров представляется более оптимальным.

Как следует из данных табл. 5, в результате программной оптимизации алгоритма Cryptonight v7 на

процессорах MALT была достигнута в 5 раз большая энергоэффективность, чем на выбранном для сравнения современном процессоре общего назначения.

Заключение

Представленные в статье результаты наглядно демонстрируют, что на процессорах MALT можно выгодно заниматься майнингом криптовалют, относящихся к типу memory-bound, и трудно переносимым на любые ASIC-устройства. Предложенные схемы оптимизации позволили улучшить энергоэффективность транзакций на MALT в 5 раз относительно процессора общего назначения. В дальнейшем авторы статьи планируют продолжать свои исследования на тему memory-bound-алгоритмов и их оптимизации для запуска на процессоре MALT.

Список литературы

1. **Елизаров С. Г., Лукьянченко Г. А., Марков Д. С.** и др. Программируемые на языках высокого уровня энергоэффективные специализированные СБИС для решения задач информационной безопасности // Системы высокой доступности. — 2018. — № 3. — С. 40–48.
2. **Cason I., Morris A., Habig B., Sun W.** Charge Me: A Comparison of Electrical Efficiency in Cryptocurrency Mining Algorithms // Issues in Information Systems. — 2018. — Vol. 19. — P. 139–149.
3. **Lamport L., Shostak R., Pease M.** The byzantine generals problem // ACM Transactions on Programming Languages and Systems. — 1982. — Vol. 4, Iss. 3. — P. 382–401.
4. **Eyal I., Sirer E. G.** Majority is not enough: Bitcoin mining is vulnerable // FC 2014: Proceedings of the International conference on financial cryptography and data security. Springer, 2014. — P. 436–454.
5. **Monero Documentation.** URL: <https://monerodocs.org/proof-of-work/cryptonight/>
6. **Tuzi D.** Cryptonight GPU Mining Efficiency, Tampere University of technology, Master of Science Thesis, 2018.
7. **Wu K., Dai G., Hu X., Li S., Xie X., Wang Y., Xie Y.** Memory-Bound Proof-of-Work Acceleration for Blockchain Applications // Proceedings of the 56th Annual Design Automation Conference. — 2019. — Article No. 177. — P. 1–6.
8. **Han R., Foutris N., Kotselidis C.** Demystifying Cryptocurrency Mining: Analysis and Optimizations of memory-hard PoW Algorithms, School of Computer Science, The University of Manchester, UK, 2019.

Таблица 5

Результаты расчета энергоэффективности

Процессор	Энергоэффективность, хеш/Вт
INTEL CORE I7-8700	3,54
XFX RADEON RX VEGA 64	5,59
ПЛИС MALT-D (RISC-only)	0,76
СБИС MALT-C-v2 (RISC + SIMD)	18,60

Transaction Accelerator for Blockchain Networks Based on Cryptonight Algorithm using Specialized Multicore Processor MALT

Titova A. A., titova_a@maltsystem.com, **V. A. Roganov**, **G. A. Lukyanenko**, **S. G. Elizarov**, Digital Microelectronics Design Center of M. V. Lomonosov Moscow State University Faculty of Physics, Moscow, 119991, Russian Federation

Corresponding author:

Titova Anna A., Researcher, Digital Microelectronics Design Center of M. V. Lomonosov Moscow State University Faculty of Physics, Moscow, 119991, Russian Federation,
E-mail: titova_a@maltsystem.com

Cryptonight is one of the possible base algorithms for cryptocurrencies. It belongs to the group of memory-bound algorithms, designed to prevent mining on specialized processors and ASICs by using 2MB of memory for each hash. Thus, it is not easy to adapt for parallel computing. The aim of this work is to prove theoretically and experimentally that this algorithm can still be optimized for a specialized multicore processor to make mining more energetically efficient than on CPU.

This article describes the process of optimization, which was conducted using the following methods: data clustering, storage of repeatedly used data in local memory, usage of SIMD for parallel computing, data prefetch. Those methods are first explained, their supposed effectiveness analyzed, and then implemented. As a result, two schemes of algorithm optimization were created: first one is based on the usage of MALT's slave cores, which compute hashes independently. Although memory-boundness creates multiple problems, we were able to increase the efficiency by clustering data. The second scheme is more complicated, it suggests using SIMD processors for most cryptographic computations and also involves data prefetch, which becomes possible if more than one hash is calculated on one core at the same time.

All the results are demonstrated in the paper and they indicate that it is indeed possible to optimize Cryptonight for a specialized multicore processor MALT. The practical results show that energy efficiency has increased 5 times in comparison with CPU.

Keywords: multicore processors, specialized processors, SIMD, cryptographic algorithms, Cryptonight, mining, memory-bound algorithms, cryptocurrencies

For citation:

Titova A. A., Roganov V. A., Lukyanchenko G. A., Elizarov S. G. Transaction Accelerator for Blockchain Networks Based on Cryptonight Algorithm using Specialized Multicore Processor MALT, *Programmnaya Ingeneria*, 2021, vol. 12, no. 6, pp. 295–301.

DOI: 10.17587/prin.12.295-301

References

1. Elizarov S., Lukyanchenko G., Markov D. et al. Energetically efficient specialized ICs programmed on high-level languages applied to problems of information security, *Sistemy vysokoj dostupnosti*, 2018, no. 3, pp. 40–48 (in Russian).
2. Cason I., Morris A., Habig B., Sun W. Charge Me: A Comparison of Electrical Efficiency in Cryptocurrency Mining Algorithms, *Issues in Information Systems*, 2018, vol. 19, pp. 139–149.
3. Lamport L., Shostak R., Pease M. The byzantine generals problem, *ACM Transactions on Programming Languages and Systems*, 1982, vol. 4, iss. 3, pp. 382–401.
4. Eyal I., Siler E. G. Majority is not enough: Bitcoin mining is vulnerable, FC-2014, *Proceedings of the International conference on financial cryptography and data security*, Springer, 2014, pp. 436–454.
5. **Monero Documentation**, available at: <https://monerodocs.org/proof-of-work/cryptonight/>
6. Tuzi D. Cryptonight GPU Mining Efficiency, Tampere University of technology, Master of Science Thesis, 2018.
7. Wu K., Dai G., Hu X., Li S., Xie X., Wang Y., Xie Y. Memory-Bound Proof-of-Work Acceleration for Blockchain Applications, *Proceedings of the 56th Annual Design Automation Conference*, 2019, article no. 177, pp. 1–6.
8. Han R., Foutris N., Kotselidis C. Demystifying Cryptomining: Analysis and Optimizations of memory-hard PoW Algorithms, School of Computer Science, The University of Manchester, UK, 2019.

О. А. Бессонов, ст. науч. сотр., bess@ipmnet.ru, Институт проблем механики им. А. Ю. Ишлинского Российской академии наук, Москва,
Н. А. Харченко, вед. специалист, Корпорация "Тактическое ракетное вооружение", Московская обл., Королёв

Программная платформа для суперкомпьютерного моделирования задач аэротермодинамики

Представлена программная платформа для суперкомпьютерного моделирования, основанная на новом технологическом подходе, существенно упрощающем распараллеливание задач по работе с неструктурированными сетками. Сформулирован принцип единого математического адресного пространства задачи для всех используемых узлов кластера. Распараллеливание кода в рамках нового подхода проводится с минимальными усилиями, без изменения основных частей программы. Представлены результаты замеров производительности, подтверждающие хорошую масштабируемость метода.

Ключевые слова: вычислительная аэротермодинамика, уравнение Навье—Стокса, неструктурированные сетки, алгоритм Катхилла—Макки, разбиение сетки, распараллеливание, OpenMP, MPI

Введение

Численное моделирование аэротермодинамических характеристик высокоскоростных летательных аппаратов требует значительных вычислительных ресурсов. Основная причина этого — необходимость в большом числе сеточных элементов (десятки или сотни миллионов ячеек) для разрешения тонких областей течения, пограничных слоев и ударных волн на гиперзвуковых режимах обтекания. Моделирование сложных мультифизических, химически неравновесных и радиационных процессов усложняет структуру числового кода и дополнительно увеличивает требования к вычислительным ресурсам [1, 2].

С распространением высокопроизводительных кластеров и суперкомпьютеров предоставляется возможность значительно ускорить процесс моделирования за счет массового распараллеливания вычислительных кодов [3, 4]. К настоящему времени накоплен большой опыт распараллеливания в моделях с общей и распределенной памятью, а также в гибридной модели OpenMP + MPI. Такая модель является наиболее перспективной для кластеров, построенных из многоядерных процессорных узлов [5]. Отработаны основные принципы распараллеливания с использованием этих моделей.

Однако распараллеливание по-прежнему остается очень сложной задачей и требует больших усилий и опыта, а также обучения программистов и координации действий между ними при создании новых кодов или адаптации старых. В связи с этим обстоятельством возрастает потребность в технологическом подходе, который мог бы упростить разработку и ускорить процесс распараллеливания существующих вычислительных кодов и создания

новых. В рамках этого подхода желательно, чтобы изменения в наследуемом последовательном коде были минимальными, а схема разбиения сетки не влияла на структуру кода. В то же время ради достижения отмеченных целей не следует сильно жертвовать эффективностью распараллеленного кода.

В предыдущей статье авторов по этой тематике [6] рассматривался подход к распараллеливанию в гибридной модели OpenMP + MPI на основе единого адресного пространства для всех подобластей параллельной программы, находящихся в разных узлах кластера. В статье был представлен пример такого распараллеливания для задачи моделирования несжимаемой вязкой жидкости в области регулярной геометрии.

Настоящая статья расширяет положения этого подхода и формулирует принцип единого математического адресного пространства в применении к задачам с неструктурированными сетками. Этот принцип удовлетворяет указанным выше требованиям как в части минимизации изменений, которые вносятся в код программы, так и простоты разбиения используемой для моделирования расчетной сетки. Данный подход реализован для задачи моделирования характеристик летательных аппаратов, однако он может быть применен и к другим задачам, использующим неструктурированные сетки с одномерной нумерацией ячеек. При этом физико-математическая специфика задачи не имеет принципиального значения.

Формализация представленного подхода и реализация его программного базиса в виде структур данных, подпрограмм обмена и правил работы с ними образуют программную платформу, на основе которой строятся параллельные расчетные программы.

1. Математическая модель и численный метод

Представленная в данной статье программная платформа использовалась для распараллеливания компьютерного кода ГРАТ [7], предназначенного для проведения расчетов аэротермодинамических характеристик высокоскоростных летательных аппаратов с применением в широком диапазоне чисел Маха, высот и углов атаки. В основе кода ГРАТ лежит решение трехмерной нестационарной системы уравнений движения вязкого, теплопроводного, химически неравновесного реагирующего газа. Система уравнений Навье—Стокса, выражающая законы сохранения массы, импульсов и полной энергии, записывается в следующем виде:

$$\begin{aligned} \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial F^x(\mathbf{w})}{\partial x} + \frac{\partial F^y(\mathbf{w})}{\partial y} + \frac{\partial F^z(\mathbf{w})}{\partial z} = \\ = \frac{\partial G^x(\mathbf{w})}{\partial x} + \frac{\partial G^y(\mathbf{w})}{\partial y} + \frac{\partial G^z(\mathbf{w})}{\partial z}, \end{aligned}$$

где $\mathbf{w} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$ — вектор консервативных переменных; ρ — плотность; u, v, w — компоненты скорости; E — полная энергия; F и G — векторные операторы конвективного и вязкого потоков.

В задачах высокоскоростного моделирования [8—11], не требующих учета физико-химических процессов в газе, решаемая система уравнений замыкается уравнениями состояния совершенного газа. В задачах гиперзвукового моделирования [12], когда рассматривается смесь химически реагирующих газов с конечными скоростями реакций, система уравнений замыкается термическим и калорическим уравнениями состояния химических компонентов газовой смеси.

Численное интегрирование конвективной части системы уравнений газовой динамики проводится с использованием метода HLLC [13], который относится к приближенным методам решения задачи о распаде произвольного разрыва, развивающим идею С. К. Годунова о вычислении потоков через грани конечного объема [14]. Для повышения порядка пространственной аппроксимации решения [15] проводится линейная реконструкция значений газодинамических параметров, используемых для вычисления конвективных потоков через грани конечного объема. Для решения системы уравнений химической кинетики используется двухшаговый метод расщепления по физическим процессам, где образование химических компонентов газовой смеси учитывается неявным образом. Вследствие того, что решение стационарной задачи проводится методом установления, используется явный метод интегрирования по времени.

Дискретизация расчетной области методом конечного объема проводится на неструктурированных сетках, важным преимуществом которых является автоматизация построения для сложных геометрических форм (рис. 1, а). Возможность построения квазирегулярной призматической сетки позволяет разрешить тонкие области течения, образующиеся в ударном слое и у поверхности обтекаемого тела на гиперзвуковых режимах (рис. 1, б). Дискретизация граничных условий основана на использовании фиктивных ячеек, прилегающих к каждой границе расчетной области.

2. Общая организация вычислительного кода ГРАТ

Вычислительный код ГРАТ написан на языке Фортран. Организация и структура вычислительного кода определяется использованием неструктурированной сетки для дискретизации вычислительной области.

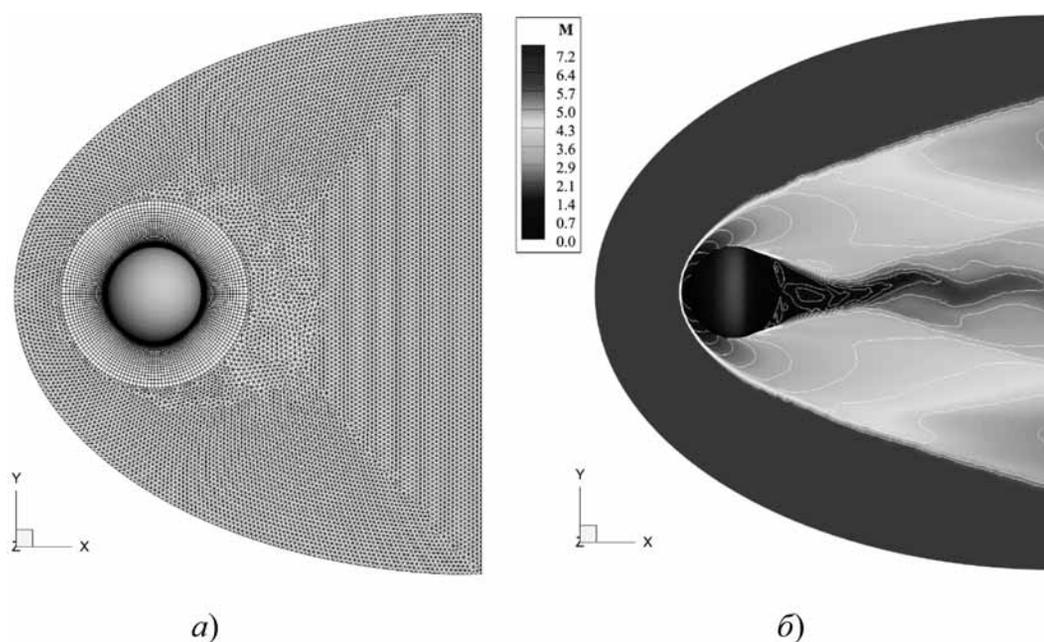


Рис. 1. Конечно-объемная сетка (а) и распределение числа Маха при гиперзвуковом обтекании затупленного тела (б)

Вычисления выполняются в соответствии с порядком нумерации ячеек (контрольных объемов). Основные структуры данных, предназначенные для хранения значений физических величин и вспомогательных переменных, имеют размерность (N) или (M, N) , где N — полное число ячеек, включая фиктивные ячейки за границей расчетной области (N_{tot}), либо число внутренних ячеек в области (N_{sum}), а M — число граней у ячейки (4 либо 5). Массивы размерности (N) используются для хранения значений величин, размещенных внутри ячейки, а размерности (M, N) — для значений, находящихся на гранях ячейки (например, для численных потоков), либо принадлежащие соседним ячейкам. Все фиктивные элементы в массивах размерности (N_{tot}) расположены в конце массивов, после внутренних элементов. Это позволяет проводить вычисления в режиме сквозного счета в DO-циклах от 1 до N_{sum} .

Для указания связей ячеек друг с другом используется массив $NB(M, N)$, в котором для каждой внутренней ячейки области хранятся номера соседних ячеек (внутренних либо фиктивных). В математическом представлении этот массив является разреженной матрицей, в каждой строке которой содержится 4 или 5 ненулевых элементов. Позиции этих элементов в строках матрицы соответствуют номерам соседних ячеек. При проведении вычислений, затрагивающих соседние ячейки, происходят обращения к элементам различных массивов по этим номерам (индексам). Алгебраически обработку таких массивов можно представить в виде $\mathbf{y} = \mathbf{A} \otimes \mathbf{x}$, где \mathbf{A} — матричное представление массива NB ; \mathbf{x} — массив, индексируемый по этой матрице; \mathbf{y} — результат вычислений и \otimes — оператор, проводящий вычисления над индексируемыми элементами массива \mathbf{x} . Такое представление аналогично записи оператора умножения разреженной матрицы на вектор.

В зависимости от способа нумерации ячеек в расчетной области ненулевые элементы в матрице \mathbf{A} могут быть размещены более или менее регулярным образом. Если такие элементы расположены максимально близко к диагонали матрицы, то индек-

сируемые элементы массивов будут сгруппированы в небольшие подмножества, помещающиеся в кеш-память. Последующие выборки элементов такого подмножества происходят из кеш-памяти, что приводит к снижению времени счета.

На практике все используемые генераторы сеток нумеруют узлы и ячейки неоптимальным образом. В связи с этим обстоятельством проводится предварительная перенумерация ячеек с использованием упрощенного варианта метода Катхилла—МакКи [16]. Благодаря такой перенумерации время счета может уменьшиться на 15...25 % (в зависимости от качества исходной сетки). Другая цель перенумерации — обеспечить эффективное разбиение сетки для кластерного распараллеливания задачи.

На рис. 2 показаны примеры портрета матрицы \mathbf{A} для исходной и перенумерованной сеток (рис. 2, а и б, соответственно). Также показана средняя часть перенумерованной матрицы в 25-кратном увеличении (рис. 2, в).

Можно заметить, что матрица \mathbf{A} после перенумерации имеет ленточную структуру. При этом ширина ленты определяет разброс адресов, по которым происходят обращения к памяти. При оптимальной перенумерации максимальная ширина ленты для трехмерной области приблизительно равна $O(N^{2/3})$.

Качество перенумерованной матрицы может зависеть от выбора ячейки, с которой начинается новая нумерация. В программном модуле применен адаптивный алгоритм с проверкой трех вариантов начальной ячейки и выбором варианта с минимальной шириной ленты.

3. Распараллеливание в модели общей памяти OpenMP

Распараллеливание в модели OpenMP проводится стандартным образом, с использованием операторов "\$OMP DO" языка Фортран. В результате применения этого оператора DO-циклы размерности N автоматически разбиваются на приблизительно равные части, которые вычисляются в параллельно исполняемых

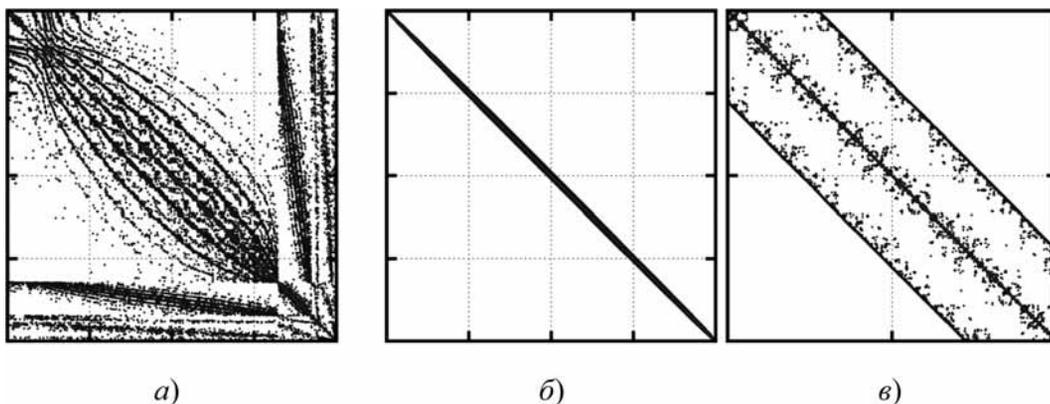


Рис. 2. Иллюстрация оптимизации сетки на примере портрета соответствующей матрицы: а — оригинальная; б — перенумерованная; в — средняя часть перенумерованной матрицы, увеличенная в 25 раз

потоках. При таком разбиении большинство обращений к индексированным элементам массивов благодаря оптимизации сетки происходит в пределах небольшого подмножества, размещенного в кеш-памяти.

Распараллеливание для систем с общей памятью, содержащих два и более процессоров, требует специальной оптимизации в связи с тем, что такие системы имеют неоднородную организацию оперативной памяти (*Non-uniform memory organization*, NUMA). Процессоры в системах типа NUMA объединены высокоскоростными каналами, и к каждому процессору подсоединен свой блок памяти. При выполнении программы обращения к данным, размещенным в памяти своего процессора, происходят с полной скоростью, а к данным в другом процессоре — с существенным замедлением. По этой причине требуется специальная инициализация основных массивов в параллельных циклах, идентичных циклам, в которых проводятся вычисления. В результате физические страницы памяти выделяются в собственной памяти соответствующего процессора.

Следует отметить, что для определенных конфигураций расчетов может оказаться предпочтительным запускать отдельный процесс MPI в каждом процессоре многопроцессорного узла вместо запуска одного процесса на узел.

Запуск расчета в системе с общей памятью должен проводиться с привязкой потоков к вычислительным ядрам. Это особенно важно для систем типа NUMA (например, двухпроцессорных узлов вычислительных кластеров), так как миграция потоков может привести к тому, что потоки и их данные окажутся в разных процессорах и скорость счета заметно снизится. В операционной системе Linux запуск расчетов с привязкой потоков проводится командой "taskset" либо "numactl", а в ОС Windows — командой "start".

4. Технология кластерного распараллеливания в модели единого математического адресного пространства

Распараллеливание задачи для вычислительных систем с распределенной памятью (кластеров) требует разбиения расчетной области на подобласти и организации расчета таким образом, чтобы вычис-

ления внутри отдельных подобластей перемежались с обменами приграничными данными.

Обмены данными между узлами кластера проводятся с помощью вызовов подпрограмм библиотеки MPI. Как правило, для такого распараллеливания создается специальный вычислительный код с организацией независимых структур данных в каждом кластерном узле и с явным вызовом обменов с соседними узлами в соответствии со структурой разбиения расчетной области [17]. При этом разбиение расчетной области означает разделение каждого исходного массива размерности (N) на подмассивы с независимой нумерацией элементов, а также подготовку специальных структур данных, описывающих границы между подобластями и стратегии обмена данными между узлами кластера.

4.1. Оптимизация и разбиение расчетной сетки

Для распараллеливания кода в представляемом программном комплексе используется технологический прием, при котором сохраняется единая сквозная нумерация ячеек и соответствующих им массивов [6]. Разбиение на подобласти проводится алгебраически путем разделения каждого исходного массива размерности (N) на подмассивы, находящиеся в разных узлах кластера. При этом используется общее пространство индексов подмассивов (единое математическое адресное пространство задачи).

Алгебраическое разбиение массивов данных означает, что в основных вычислениях, которые имеют вид $y = A \otimes x$, на границах между подмассивами будет наблюдаться перекрытие. На рис. 3 показаны шаблоны доступа к данным при вычислении $A \otimes x \rightarrow y$ внутри подмассива (рис. 3, а) и вблизи границы между подмассивами (рис. 3, б).

Можно видеть, что для каждого массива, к которому осуществляется доступ как к вектору x , при выделении памяти для подмассивов должно быть дополнительно добавлено место для области перекрытия. Размер каждой области перекрытия примерно равен полуширине ленты матрицы A в этой позиции. После каждого вычисления вида $y = A \otimes x$ должен быть проведен обмен перекрывающимися областями некоторых массивов между соответствующими узлами кластера. Для экономии времени счета обмен

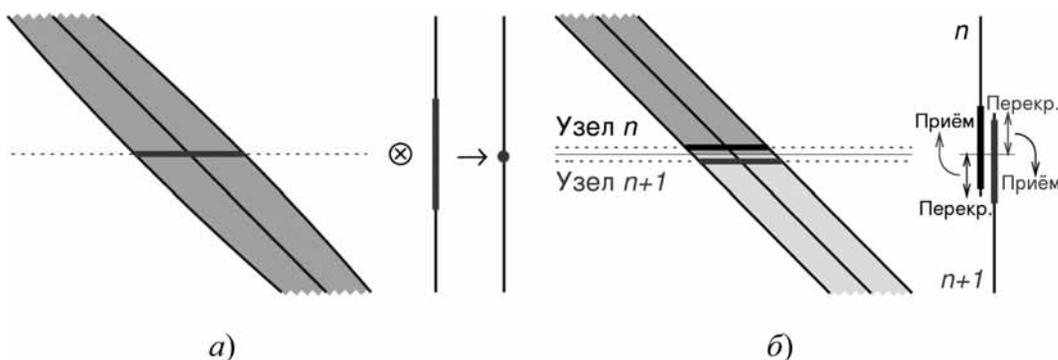


Рис. 3. Иллюстрация характера доступа к данным:

а — внутри подмассива, без перекрытия; б — между подмассивами, находящимися в разных узлах кластера, с перекрытием адресов и обменом данными

данными в противоположных направлениях должен проводиться одновременно (в дуплексном режиме).

В текущей реализации компьютерного кода используются синхронные режимы обмена приграничными данными, что при разбиении на относительно небольшое число подобластей не приводит к существенным накладным расходам. Использование асинхронных режимов обмена данными, происходящих одновременно с вычислениями, потребует более сложных способов обхода массивов и будет реализовано позднее, по мере получения доступа к кластерам с большим числом узлов.

Алгебраический метод разбиения, будучи простым и понятным, не является самым оптимальным. Существует ряд сложных методов декомпозиции доменов, которые минимизируют размер граничных областей между подобластями [18–20]. На рис. 4 приведены примеры почти оптимального разбиения с использованием таких методов.

Однако сложные методы декомпозиции, близкие к оптимальным, имеют ряд недостатков. Как правило, подобласти после такого разбиения имеют большое (и различное) число соседних подобластей с неравными размерами границ. В результате схемы обмена становятся сложными и запутанными, что

усложняет алгоритм распараллеливания и затрудняет его оптимальную программную реализацию. По этой причине применение таких методов оправдано только при разбиении на большое число подобластей.

Вместе с тем при использовании алгебраического метода разбиения каждая подобласть имеет только двух соседей, и размеры границ между подобластями не сильно отличаются. Эти свойства частично компенсируют неоптимальный характер данного метода.

При использовании метода Катхилла—Макки ячейки нумеруются слой за слоем, и характер распространения фронтального слоя по расчетной области зависит от вариаций плотности сетки. При наличии сильных сжатий сетки фронтальный слой может в этих местах искривляться и общее число ячеек в слое может увеличиваться. В местах разбиения расчетной области это значение определяет число граничных ячеек между подобластями и, соответственно, полуширину ленты матрицы A , которая соответствует объему обменов данными между узлами кластера.

На рис. 5 показаны примеры разбиения трехмерных расчетных областей на восемь подобластей с использованием этого метода. На этих примерах можно видеть искривление границ между подобластями. Число ячеек сетки (N) и полуширина ленты матрицы (H) для этих примеров следующие: $N = 12$ млн, $H = 78\,000$ (рис. 5, а); $N = 6$ млн, $H = 28\,000$ (рис. 5, б). Значения параметра $H/N^{2/3}$, указывающие относительное (нормализованное к размеру сетки) значение полуширины матрицы, составляют 1,48 и 0,84 соответственно. Это означает, что оба разбиения сетки имеют приемлемое качество, при этом качество второго разбиения немного лучше, чем первого.

4.2. Структура вычислительного кода

Технически расчеты в подобластях организованы следующим образом. Для всех подмассивов данных, совместно использующих единое индексное пространство задачи, определены соответствующие целочисленные переменные, обозначающие диапазоны размещения внутренних (реальных) ячеек,

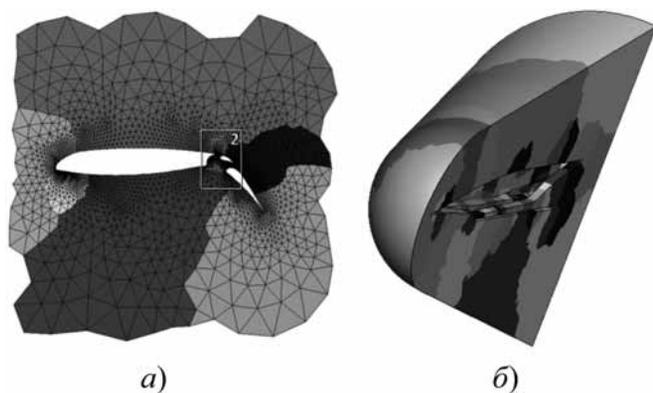


Рис. 4. Иллюстрация почти оптимального разбиения области на 8 подобластей в двумерном случае [20] (а) и на 64 подобласти в трехмерном [17] (б)

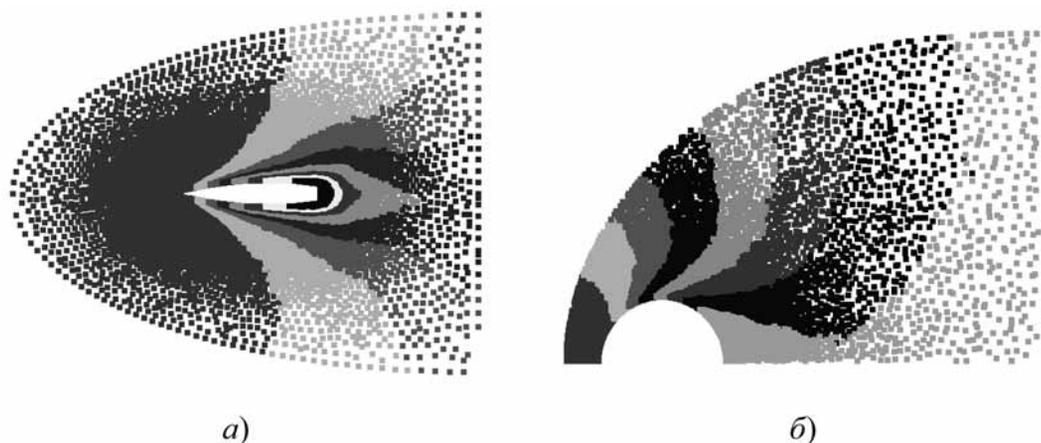


Рис. 5. Примеры применения алгебраического метода разбиения сетки

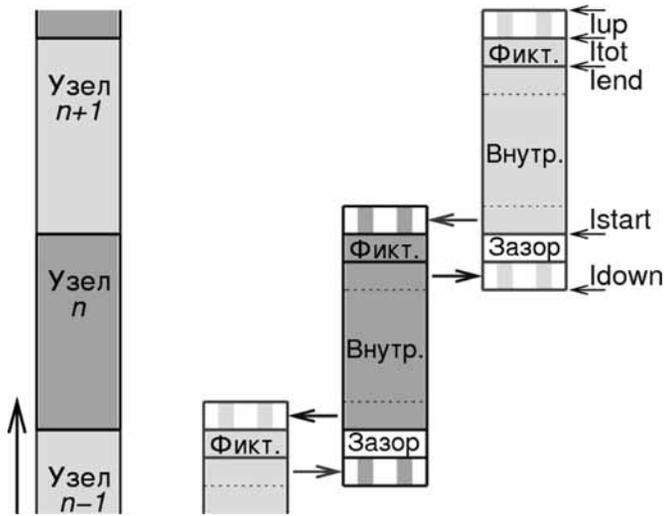


Рис. 6. Структура подмассива в едином пространстве индексов задачи и иллюстрация обменов между перекрывающимися областями

фиктивных ячеек и областей перекрытия. На рис. 6 показана структура подмассива и характер данных, которыми обмениваются подобласти.

Основными диапазонами индексных переменных являются $Istart$: $Iend$ для обработки DO-циклов и $Idown$: Iup для выделения памяти для подмассивов. Индексные пространства подмассивов в соседних узлах кластера перекрываются, при этом соотношение между локальными индексными переменными выглядит следующим образом: значение $Itot + 1$ для узла n должно быть равно значению $Istart$ для узла $n + 1$.

Вычисления в узлах кластера выполняются с распараллеливанием в модели OpenMP. После завершения каждого этапа вычислений проводится обмен с обоими соседними узлами. Используется процедура `MPI_SENDRECV`, которая одновременно принимает данные от соседнего узла и передает данные этому узлу.

Технологически все процедуры обмена данными отделены от вычислительного кода и размещаются в специальном файле `XCH.F`. Все основные обмены

между соседними подобластями реализованы в подпрограмме `XCH`, которая вызывается один раз после каждого этапа вычислений (для каждого вновь вычисленного массива, имеющего перекрывающиеся области). Эта подпрограмма самостоятельно определяет, исходя из номера текущего узла и их общего числа, с какими подобластями и в каком порядке проводить операции встречных обменов. В случае, если задача запущена на единственном узле, такая подпрограмма не будет выполнять никакой работы.

На рис. 7, а, показано, как работает подпрограмма `XCH`. Подпрограмма делает два вызова `MPI_SENDRECV` в соответствующем порядке для обмена с обоими соседними узлами кластера.

Аналогичным образом реализованы подпрограммы редукции для вычисления суммы значений переменных во всех узлах, а также их максимального и минимального значений (`XSUM`, `XMAX`, `XMIN`). Для этих вычислений используется алгоритм типа "бабочка" (подобно применяемому в быстром преобразовании Фурье) (рис. 7, б).

Также в коде реализованы унифицированные подпрограммы для распределения данных между узлами кластера после ввода и для сбора данных из узлов перед выводом.

Выделение всех вызовов подпрограмм библиотеки `MPI` в отдельный файл позволяет компилировать основной код обычным компилятором. Компилятор, поддерживающий `MPI`, необходим только для окончательной сборки с использованием этого файла. Также программа может быть собрана без библиотеки `MPI` — для этого случая реализован набор подпрограмм, имитирующих выполнение на одном узле без обменов (`XCH_NOMPI.F`).

Рисунок 8 иллюстрирует эволюцию компьютерного кода в виде этапов — чисто последовательного кода, параллельного кода в модели `OpenMP` и параллельного кода в кластерной модели. На рис. 8 выделены принципиальные изменения основного `DO`-цикла на каждом этапе развития: добавление операторов `!$OMP` для распараллеливания в `OpenMP`, вызов универсальной подпрограммы обмена `XCH` и коррекция диапазона `DO`-цикла для кластерного распараллеливания.

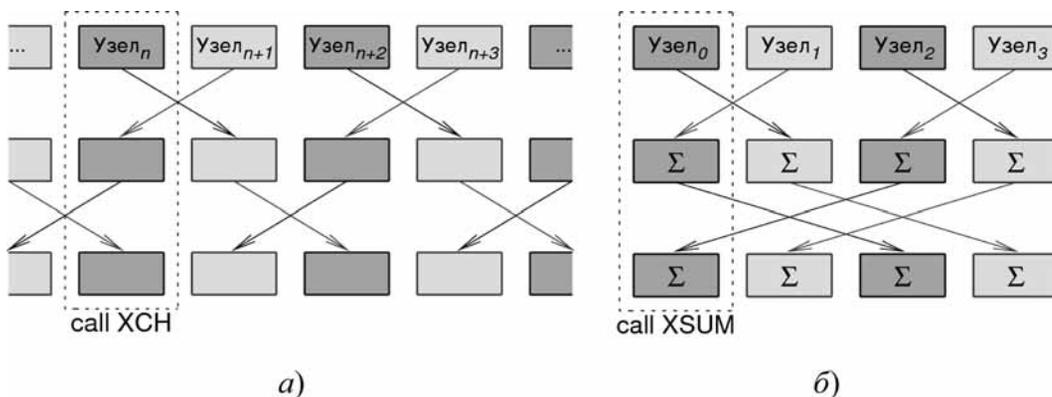


Рис. 7. Иллюстрация основных типов обменов между узлами кластера:

а — обмен с соседними узлами по завершении этапа вычислений; б — алгоритм обменов логарифмической сложности для операций редукции

```

do I=1,Nsum
...
A(I)= ...
B(K,I)= ...
...
enddo

```

a)

```

!$OMP DO
do I=1,Nsum
...
A(I)= ...
B(K,I)= ...
...
enddo
!$OMP END DO

```

б)

```

!$OMP DO
do I=Istart,Iend
...
A(I)= ...
B(K,I)= ...
...
enddo
!$OMP END DO
!$OMP SINGLE
call XCH(...,A)
call XCH(...,B)
!$OMP END SINGLE

```

в)

Рис. 8. Эволюция вычислительного кода:
a — последовательный код; б — OpenMP; в — OpenMP + MPI

Таким образом, технология кластерного распараллеливания с единым математическим адресным пространством задачи позволяет реализовать унифицированный расчетный код для всех режимов — однопоточного, многопоточного в модели общей памяти OpenMP, и кластерного в гибридной модели OpenMP + MPI. При этом распараллеливание кода для кластерного режима обеспечивается минимальными усилиями, без переделки основных частей программы, и при единственном условии — наличии естественного потенциала распараллеливания (например, при использовании явного метода интегрирования по времени). Метод распараллеливания обеспечивает достаточно хорошее качество разбиения расчетной области, при котором число граничных ячеек между подобластями не превышает уровня $O(N^{2/3})$.

5. Измерения производительности

Новый алгоритм распараллеливания был протестирован на кластере, построенном из двухпроцессорных узлов на процессорах Intel Xeon, содержащих 16 ядер с тактовой частотой 2,6...2,9 ГГц. Узлы кластера соединены между собой каналами Omni-Path со скоростью обмена 100 Гбит/с. Результаты распараллеливания при решении задачи аэротермодинамики с сеткой, содержащей 33 млн ячеек, показаны на рис. 9.

На рис. 9, а, показано ускорение при включенном режиме Hyper-threading и запуске одного процесса MPI на узел в режиме NUMA. В этом сочетании режимов эффект насыщения параллельной производительности начинает наблюдаться на 24 узлах (768 процессорных ядер, 1536 потоков). График на рис. 9, б, соответствует отключенному режиму Hyper-threading и запуску отдельного процесса MPI в каждом процессоре. Здесь насыщение достигается на 32 узлах (1024 ядра). Данное сравнение показывает, что при разбиении на большое число подобластей начинают проявляться эффекты вследствие замедленного доступа к нелокальной памяти в режиме NUMA.

В таблице показаны ускорение и эффективность распараллеливания задачи для этих режимов (режим 1 — один процесс на узел, Hyper-threading включен; режим 2 — два процесса на узел, Hyper-threading отключен).

Приведенные выше результаты демонстрируют хорошую масштабируемость метода распараллеливания

Ускорение и эффективность распараллеливания задачи

Число узлов	Режим 1		Режим 2	
	Ускорение	Эффективность, %	Ускорение	Эффективность, %
1	1	100	1	100
2	1,95	97,4	1,97	98,6
4	3,79	94,7	3,83	95,6
6	5,54	92,3	5,64	94,1
8	7,18	89,8	7,34	91,8
12	10,4	86,4	10,6	88,2
16	13,1	83,1	13,3	82,9
24	16,4	68,2	19,3	80,3
28	—	—	21,9	77,8
32	—	—	22,9	71,5

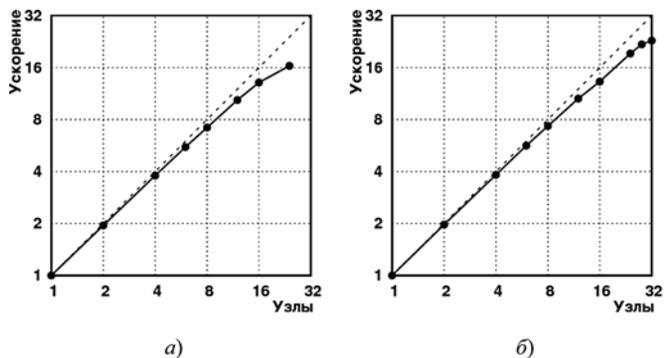


Рис. 9. Ускорение при распараллеливании задачи с сеткой, содержащей 33 млн ячеек, в различных сочетаниях режимов Hyper-threading и MPI

ния. Основываясь на 23-кратном ускорении на одном узле кластера по сравнению с одним потоком, общее ускорение на 32 узлах составляет 520. Таким образом, эти результаты иллюстрируют широкий спектр вычислительных платформ, на которых метод может эффективно использоваться.

Заключение

Представлен новый технологический подход к реализации эффективного кластерного распараллеливания газодинамических и аэродинамических задач, использующих неструктурированные сетки, и программная платформа, построенная на его основе. Этот подход основан на едином математическом адресном пространстве данных задачи для всех используемых узлов кластера задачи. Благодаря рассматриваемому подходу распараллеливание кода осуществляется с минимальными усилиями, без изменения основных частей программы. Кроме того, такой подход значительно упрощает разбиение расчетной сетки.

В статье также представлены приемы, упрощающие организацию обменов между узлами кластера и облегчающие программирование. В частности, все процедуры обмена между узлами, содержащие вызовы подпрограмм библиотеки MPI, отделены от основного кода.

В результате применения описанных выше подходов создается единый вычислительный код для всех режимов счета — последовательного, многопоточного (в модели общей памяти OpenMP) и кластерного (в модели распределенной памяти OpenMP + MPI).

Представленный подход может быть применен к различным задачам, использующим неструктурированные сетки с одномерной нумерацией ячеек и допускающим распараллеливание на основе разбиения расчетной области.

Разработанная программная платформа использовалась для распараллеливания компьютерного кода ГРАТ. Результаты измерений производительности кода демонстрируют хорошую масштабируемость предложенного метода распараллеливания.

Работа выполнена при частичной поддержке из средств госбюджета в рамках государственного задания АААА-А20-120011690131-7.

Список литературы

1. Shang J. S., Surzhikov S. T. Simulating nonequilibrium flow for ablative Earth reentry // *J. Spacecraft Rockets*. — 2010. — Vol. 47. — P. 806–815.
2. Zheleznyakova A. L., Surzhikov S. T. Application of the method of splitting by physical processes for the computation of a hypersonic flow over an aircraft model of complex configuration // *High Temp*. — 2013. — Vol. 51. — P. 816–829.
3. Afzal A., Ansari Z., Faizabadi A. R., Ramis M. K. Parallelization strategies for computational fluid dynamics software:

state of the art review // *Arch. Computat. Methods Eng.* — 2017. — Vol. 24. — P. 337–363.

4. Probst A., Knopp T., Grabe C., Jägersküpper J. HPC requirements of high-fidelity flow simulations for aerodynamic applications // *In Proc. Euro-Par 2019, Lecture Notes in Computer Science*. — 2019. — Vol. 11997. — P. 375–387.

5. Rabenseifner R., Hager G., Jost G. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes // *In 17th Euromicro international conference on parallel, distributed and network-based processing*. — 2009. — P. 427–436.

6. Bessonov O. Technological aspects of the hybrid parallelization with OpenMP and MPI // *In Proc. PaCT 2017, Lecture Notes in Computer Science*. — 2017. — Vol. 10421. — P. 101–113.

7. Харченко Н. А., Рыбаков А. Н. Аэротермодинамика спускаемого космического аппарата "Аполлон-4" // *Точно в цель*. — 2020. — № 1. — С. 77–88.

8. Kharchenko N., Kryukov I. Aerothermodynamics calculation of the EXPERT reentry flight vehicle // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012004.

9. Kharchenko N., Kotov M. Analysis of the High Speed Gas Flow over a Sphere in the Range of Mach Numbers 2–12 // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012007.

10. Koryukov I., Kryukov I. Three-dimensional calculation of the aerothermodynamics of a double cone 25°/55° on an unstructured grid // *J. Phys.: Conf. Ser.* — 2018. — Vol. 1009. — Article 012003.

11. Макенч Г. С., Харченко Н. А., Крюков И. А. Расчет аэродинамики и динамики полета спускаемого летательного аппарата EXPERT // *Физико-химическая кинетика в газовой динамике*. — 2017. — Т. 18, Вып. 2. URL: <http://chemphys.edu.ru/issues/2017-18-2/articles/738/>

12. Kharchenko N., Kotov M. Aerothermodynamics of the Apollo-4 spacecraft at earth atmosphere conditions with speed more than 10 km/s // *J. Phys.: Conf. Ser.* — 2019. — Vol. 1250. — Article 012012.

13. Einfeldt B., Munz C. D., Roe P. L., Sjögreen B. On Godunov-type methods near low densities // *J. Comput. Phys.* — 1991. — Vol. 92. — P. 273–295.

14. Годунов С. К. Разностный метод численного расчета разрывных решений уравнений гидродинамики // *Матем. сб.* — 1959. — Т. 47 (89), № 3. — С. 271–306.

15. Крюков И. А., Иванов И. Э., Ларина Е. В. Программный комплекс расчета высокоскоростных течений hySol // *Физико-химическая кинетика в газовой динамике*. — 2021. — Т. 22, Вып. 1. URL: <http://chemphys.edu.ru/issues/2021-22-1/articles/902/>

16. Cuthill E., McKee J. Reducing the bandwidth of sparse symmetric matrices // *In ACM'69: Proc. 24th Nat. Conf.* — 1969. — P. 157–172.

17. Ermakov M., Kryukov I. Supercomputer modeling of flow past hypersonic flight vehicles // *J. Phys.: Conf. Ser.* — 2017. — Vol. 815. — Article 012016.

18. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs // *SIAM J. Sci. Comput.* — 1998. — Vol. 20, No. 1. — P. 359–392.

19. Karypis G., Kumar V. Multilevel κ -way partitioning scheme for irregular graphs // *J. Parallel Distrib. Comput.* — 1998. — Vol. 48. — P. 96–129.

20. Железнякова А. Л. Эффективные методы декомпозиции неструктурированных адаптивных сеток для высокопроизводительных расчетов при решении задач вычислительной аэродинамики // *Физико-химическая кинетика в газовой динамике*. — 2017. — Т. 18, Вып. 1. URL: <http://chemphys.edu.ru/issues/2017-18-1/articles/673/>

Software Platform for Supercomputer Modeling of Aerothermodynamics Problems

O. A. Bessonov, bess@ipmnet.ru, Ishlinsky Institute for Problems in Mechanics RAS, Moscow, 119526, Russian Federation, N. A. Kharchenko, kharchenko.na@phystech.edu, Tactical Missiles Corporation, Moscow district, Korolev, 141080, Russian Federation

Corresponding author:

Bessonov Oleg A., Senior Research Scientist, Ishlinsky Institute for Problems in Mechanics RAS, Moscow, 119526, Russian Federation
E-mail: bess@ipmnet.ru

Received on April 15, 2021

Accepted on June 21, 2021

Numerical solution of aerothermodynamics problems requires a lot of processor time, and for this reason, the numerical codes for such simulations must be efficiently parallelized. This paper presents a software platform based on a technological approach that greatly simplifies the parallelization of problems with unstructured grids. The paper formulates the principle of a unified mathematical address space of the problem for all used cluster nodes. The formalization of the presented technological approach and the implementation of its software basis in the form of data structures, exchange routines and work rules form a software platform on the basis of which parallel computational programs can be built. This approach is implemented and tested for the problem of modeling the characteristics of aircrafts, but it can be applied to other problems using unstructured grids with one-dimensional cell numbering. In this case, the physical and mathematical specifics of the problem are not important. Parallelization of the code with the new approach is carried out with minimal effort, without changing the main parts of the program. As a result, a single computational code can be created for all regimes — sequential, multi-threaded, and cluster. The performance results obtained with the new code confirm the good scalability of the parallelization method.

Keywords: computational aerothermodynamics, Navier—Stokes equation, unstructured grids, Cuthill—McKee algorithm, unified mathematical address space, grid partitioning, parallelization, OpenMP, MPI, performance evaluation

Acknowledgements:

This work was partially supported by the Russian State Assignment under Contract No. AAAA-A20-120011690131-7.

For citation:

Bessonov O. A., Kharchenko N. A. Software Platform for Supercomputer Modeling of Aerothermodynamics Problems, *Programmnyaya Ingeneriya*, 2021, vol. 12, no. 6, pp. 302—310.

DOI: 10.17587/prin.12.302-310

References

1. Shang J. S., Surzhikov S. T. Simulating nonequilibrium flow for ablative Earth reentry, *J. Spacecraft Rockets*, 2010, vol. 47, pp. 806—815.
2. Zheleznyakova A. L., Surzhikov S. T. Application of the method of splitting by physical processes for the computation of a hypersonic flow over an aircraft model of complex configuration, *High Temp.*, 2013, vol. 51, pp. 816—829.
3. Afzal A., Ansari Z., Faizabadi A. R., Ramis M. K. Parallelization strategies for computational fluid dynamics software: state of the art review, *Arch. Computat. Methods Eng.*, 2017, vol. 24, pp. 337—363.
4. Probst A., Knopp T., Grabe C., Jägersküpper J. HPC requirements of high-fidelity flow simulations for aerodynamic applications, *In Proc. Euro-Par 2019, Lecture Notes in Computer Science*, 2019, vol. 11997, pp. 375—387.
5. Rabenseifner R., Hager G., Jost G. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes, *In 17th Euromicro international conference on parallel, distributed and network-based processing*, 2009, pp. 427—436.
6. Bessonov O. Technological aspects of the hybrid parallelization with OpenMP and MPI, *In Proc. PaCT 2017, Lecture Notes in Computer Science*, 2017, vol. 10421, pp. 101—113.
7. Kharchenko N. A., Rybakov A. N. Aerothermodynamics of the Apollo-4 re-entry vehicle, *Tochno v cel'*, 2020, no. 1, pp. 77—88 (in Russian).
8. Kharchenko N., Kryukov I. Aerothermodynamics calculation of the EXPERT reentry flight vehicle, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, 012004.
9. Kharchenko N., Kotov M. Analysis of the High Speed Gas Flow over a Sphere in the Range of Mach Numbers 2—12, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, article 012007.
10. Koryukov I., Kryukov I. Three-dimensional calculation of the aerothermodynamics of a double cone 25°/55° on an unstructured grid, *J. Phys.: Conf. Ser.*, 2018, vol. 1009, article 012003.
11. Makeich G. S., Kharchenko N. A., Kryukov I. A. Aerodynamics and flight dynamics simulation of EXPERT re-entry vehicle, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2017, vol. 18, no. 2, available at: <http://chemphys.edu.ru/issues/2017-18-2/articles/738/> (in Russian).
12. Kharchenko N., Kotov M. Aerothermodynamics of the Apollo-4 spacecraft at earth atmosphere conditions with speed more than 10 km/s, *J. Phys.: Conf. Ser.*, 2019, vol. 1250, article 012012.
13. Einfeldt B., Munz C. D., Roe P. L., Sjögren B. On Godunov-type methods near low densities, *J. Comput. Phys.*, 1991, vol. 92, pp. 273—295.
14. Godunov S. K. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, *Matematicheskij sbornik*, 1959, vol. 47 (89), no. 3, pp. 271—306 (in Russian).
15. Kryukov I. A., Ivanov I. E., Larina E. V. Software package hySol for the Numerical Simulation of High-speed Flows, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2021, vol. 22, no. 1, available at: <http://chemphys.edu.ru/issues/2021-22-1/articles/902/> (in Russian).
16. Cuthill E., McKee J. Reducing the bandwidth of sparse symmetric matrices, *In ACM'69: Proc. 24th Nat. Conf.*, 1969, pp. 157—172.
17. Ermakov M., Kryukov I. Supercomputer modeling of flow past hypersonic flight vehicles, *J. Phys.: Conf. Ser.*, 2017, vol. 815, article 012016.
18. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.*, 1998, vol. 20, no. 1, pp. 359—392.
19. Karypis G., Kumar V. Multilevel κ -way partitioning scheme for irregular graphs, *J. Parallel Distrib. Comput.*, 1998, vol. 48, pp. 96—129.
20. Zheleznyakova A. L. Effective domain decomposition methods for adaptive unstructured grids applied to high performance computing for problems in computational aerodynamics, *Fiziko-khimicheskaya kinetika v gazovoy dinamike*, 2017, vol. 18, no. 1, available at: <http://chemphys.edu.ru/issues/2017-18-1/articles/673/> (in Russian).

Д. С. Пащенко, канд. техн. наук, MBA, denpas@ Rambler.ru, независимый консультант в области разработки программного обеспечения

Российский опыт организации полностью удаленной разработки программного обеспечения: отраслевое исследование 2021 г.

Пандемия, и как следствие, появление новых подходов к управлению во многих сферах деятельности человека продолжают оказывать значительное влияние на развитие новой COVID-экономики. Такое влияние наблюдается в ускорении процессов цифровизации, виртуализации рабочих процессов, преобладании коммуникаций с использованием электронных каналов. Переход к полностью удаленной разработке и поддержке программного обеспечения (ПО) под влиянием COVID-фактора не просто изменил IT-отрасль, а ознаменовал появление нового стандарта организации производства. В статье приведены результаты исследования 2021 г., охватившего различные аспекты удаленной разработки ПО в российских технологических компаниях, включая ВТБ, Яндекс, Сбертех и др. Более 20 экспертов, представляющих свои команды разработчиков, определили основные подходы к обеспечению мотивации и к вовлечению инженеров, а также к обеспечению благоприятных условий их профессионального роста. Также в исследовании определены перспективы развития полностью удаленной разработки ПО как нового отраслевого стандарта. Результаты исследования сопровождаются краткими выводами, основная идея которых в необходимости быстрого осознания нового тренда в разработке ПО и использовании данного фактора в конкурентной борьбе (наем специалистов, организация команд, организационное развитие).

Ключевые слова: удаленная работа, информационные технологии, разработка ПО, COVID-экономика, технологии ПО

Введение

Пандемия COVID-19 и обусловленные ею волны социальной изоляции (локдауны) существенно изменили организацию труда в отрасли информационных технологий, в частности, в командах, разрабатывающих программное обеспечение (ПО). Исследование 2020 г. [1] показало, что мировые и европейские IT-лидеры довольно легко перешли к практике полностью удаленной разработки ПО вне офисов, используя накопленный опыт географически распределенных команд и соответствующей автоматизации рабочих процессов. Здесь и далее — полностью удаленная разработка (*fully remote development*) — современный паттерн разработки ПО, возникший в 2020 г. и подразумевающий работу абсолютно всей проектной команды вне офиса компании-разработчика. Значительную положительную роль в таком режиме разработки ПО сыграло использование виртуализации рабочих процессов, ставшее частью современных подходов в управлении качеством ПО. К таким процессам относятся: управление требованиями, контроль версионности, непрерывная интеграция и поставка.

Ускорение цифровой трансформации под воздействием пандемии [2], а значит возрастающий спрос на ПО,

стали позитивным фактором. Однако удовлетворение такого растущего спроса требует частичного перестроения всех процессов организации производства ПО. Несмотря на быстрый переход к полностью удаленной разработке и первоначальное уверенное впечатление экспертов отрасли о поддержании стабильности ключевых показателей эффективности в IT-компаниях (долгосрочном качестве ПО, продуктивности команд и т. п.) [1], с течением времени вопросы долгосрочной мотивации отдельных разработчиков и вовлеченности команд в целом в проектные вопросы стали более заметными. Все больше исследователей в мире [3–5] анализируют возникшие сложности, в том числе связанные со смещением времени и пространства между работой разработчика ПО и членов его семьи, необходимостью отвлекаться на рутинные домашние проблемы, де-социализацией сотрудников и т. п.

В частности, отмечалось, что радикальное снижение неформального общения в коллективе негативно влияет на вовлеченность разработчиков в решение бизнес- и технических задач, поставленных заказчиками и потенциальными пользователями создаваемого ПО. Оно затрудняет привычные модели профессионального роста разработчиков, связанные с наставничеством и обменом опытом.

Перенос важнейших рабочих процессов в онлайн-формат в любой парадигме разработки ПО (Scrum, RUP, MSF) вынуждает руководство компании (и проектов) к пересмотру методов долгосрочной мотивации специалистов, участвующих в разработке, и меняет принципы командообразования. Ключевые вопросы и задачи, решение которых нуждается в модернизации в условиях пандемии и полностью удаленной командной разработки ПО:

- долгосрочная высокая мотивация и вовлеченность разработчиков в решение бизнес-задач и удовлетворение требований заказчиков и потенциальных клиентов разрабатываемого ПО;
- профессиональный рост младших специалистов в команде разработчиков;
- рост управленческих и профессиональных компетенций опытных специалистов-разработчиков, в перспективе стремящихся стать лидерами команд разработки, проектными менеджерами и системными архитекторами.

В 2021 г. становится очевидно, что эти вопросы носят глобальный характер и будут оставаться актуальными как минимум в среднесрочной перспективе. Следует ожидать, что даже после завершения пандемии значительная часть IT-компаний сохранит практики полностью удаленной разработки ПО или "гибридные" модели [1], предпочитая разрешать обозначенные выше вопросы вместо полномасштабного возвращения сотрудников в офисы. Тем более что часть мировых лидеров IT-отрасли уже изменила модели найма инженеров, которые теперь не связаны ни с их географической локацией, ни с возможностями релокации нанимаемых сотрудников.

Таким образом, автором были определены гипотезы, связанные с отмеченными выше ключевыми вопросами. В **гипотезе 1** предполагается, что в условиях полностью удаленной разработки руководители проектов, IT-департаментов, софтверных компаний разрабатывают и апробируют новые методы и подходы для краткосрочной и долгосрочной мотивации специалистов-разработчиков и их вовлечения в решение бизнес-задач и удовлетворение требований пользователей и заказчиков разрабатываемого ПО.

Традиционные модели мотивации инженеров в IT-отрасли связаны, в основном, с материальными поощрениями, возможностями профессионального роста, использования передовых и уникальных технологий, фокусом на создание выдающихся решений, изменяющих окружающий мир [6–8]. Гипотеза 1 призвана ответить на вопрос, как в новых условиях традиционные модели краткосрочной и долгосрочной мотивации изменяются в российских командах, разрабатывающих ПО и информационные сервисы?

В **гипотезе 2** в контексте рассматриваемого исследования автор предполагает, что в условиях полностью удаленной работы руководством проектов и компаний используются какие-то новые методы и подходы для обеспечения благоприятных условий для профессионального и личного роста специалистов-разработчиков. Причем такие уси-

лия руководителей затрагивают как начинающих, так и опытных разработчиков и аналитиков, стремящихся сделать следующий шаг в своей карьере и стать лидером команды или проектным менеджером.

Типовые приемы наставничества и курирования новичков, обучение через постановку задач в режиме "растягивание/зона комфорта" и другие эффективные приемы [9, 10] очевидно должны были претерпеть изменения, направленные на увеличение их эффективности в условиях полностью удаленной работы. Гипотеза 2 предполагает, что накопление таких подвижек неизбежно приведет к качественному изменению подходов, методов и средств, способствующих профессиональному росту разработчиков в новых условиях.

В **гипотезе 3** в данном исследовании сделано предположение, что в подавляющем большинстве компаний и команд в софтверной разработке уже определены эффективные механизмы работы в условиях полностью удаленной разработки ПО. В рамках проверки данной гипотезы интервьюируемым экспертам также было предложено оценить возможные пути улучшения удачных организационных паттернов в полностью удаленной разработке. В статье проведено сравнение полученных результатов при проверке гипотезы 3 с результатами исследования [1], проведенного в 2020 г.

Цель, задачи и метод исследования

Анализ успешных практик долгосрочной мотивации разработчиков ПО и командообразования для работы в условиях удаленной разработки программного продукта является основной целью данной работы. Предложенные выше гипотезы согласовывались с экспертами, представляющими российские IT-компании и проектные команды. В рамках исследования также изучались уже опробованные на практике организационные методы. По результатам такого анализа было сформировано коллективно согласованное экспертное мнение специалистов о перспективных подходах в дистанционной совместной деятельности по разработке ПО.

Более ранние авторские исследования [11, 12] показали, что российская IT-отрасль является очень восприимчивой к инновациям, например, к лидирующим в мире технологиям и паттернам организации командной работы. Исследования, результаты которых представлены в настоящей статье, направлены на оценку скорости усвоения российскими командами аналогичных инноваций, применительно к совершенствованию практик удаленной разработки ПО. Результаты предыдущих исследований на этом направлении позволяют предположить, что уровень встраивания российских IT-команд в мировые тенденции достаточно высок. Это предопределяет заметную схожесть в стремлении улучшать практики удаленной разработки, ставшие для России "вынужденными", а для ведущих мировых лидеров — уже "осознанным" стандартом организации производства ПО и информационных сервисов.

Среди частных задач исследования следует отметить упомянутые ранее:

1) определение успешных методов краткосрочной и долгосрочной мотивации проектных команд и увеличения вовлеченности программистов в бизнес-вопросы, связанные с создаваемым ПО;

2) поиск возможностей обеспечения условий для профессионального роста начинающих программистов в командах разработки, функционирующих в удаленном режиме;

3) создание новых подходов к обеспечению благоприятных возможностей роста профессиональных и управленческих компетенций для опытных разработчиков в новых условиях.

Решение частных задач направлено на достижение поставленной цели и проверку адекватности положений трех обозначенных ранее гипотез. Оно также включает в себя анализ уровня успешности уже применяемых методов в IT-компаниях и организациях, в которых разработка ПО обеспечивает значительные преимущества в отраслевой конкурентной борьбе (банки, страховые организации, телеком и т. п.).

В качестве метода исследования были выбраны дистанционные интервью с обязательным набором открытых вопросов по гипотезам, задачам и по цели исследования. После завершения всех интервью экспертам были высланы обобщенные результаты, которые можно было дополнить в части своих ответов и прокомментировать полученные результаты. Всего в исследовании в январе—марте 2021 г. приняло участие 22 эксперта, представивших опыт разработки ПО в 2020 г. в российских IT-компаниях, банках, цифровых рекламных агентствах (включая ВТБ, Яндекс, СберТех).

Эксперты приглашались к сотрудничеству рассылкой синопсиса исследования в различных электронных каналах (LinkedIn, email, telegram). Дополнительные данные по экспертам представлены в таблице.

Эксперты в исследовании

Эксперты	IT-компании	Компании других отраслей
С ролью программистов	9	3
С ролью руководителя	7	3

Выбранный метод исследования подразумевает обобщение полученных мнений экспертов при проверке гипотез и решении поставленных задач. Анализ собранных перспективных приемов, улучшающих практику полностью удаленной разработки, приведен в заключительной части следующего раздела.

Основные результаты

Краткосрочная и долгосрочная мотивация, вовлечение специалистов-разработчиков в бизнес-вопросы заказчиков и пользователей

Успех деятельности команд разработки ПО зависит не только от зрелости процессов и технологической оснащенности, но и от общего уровня

мотивации специалистов, участвующих в разработке. Краткосрочная мотивация определяет поведение таких специалистов в ближайшем будущем. Долгосрочная мотивация имеет целью формирование мотивов, действующих на продолжительную перспективу. Краткосрочная мотивация при этом, как правило, основана на применении традиционных, но действенных стимулов. К их числу относятся, например, такие как премирование, благодарность, выделение заслуг и вклада в общий результат, определение лучших работников и т. п. Управление мотивацией, формируемой на длительную перспективу, ориентировано на воздействия, прямо влияющие на жизненные ориентиры, ценности и цели разработчиков ПО, связанные с профессиональным развитием, повышением стоимости его услуг на рынке, освоением новых технологий.

Проблема эффективного и рассчитанного на длительную перспективу вовлечения специалистов-разработчиков в разрешение вопросов заказчиков и пользователей разрабатываемого ПО остается актуальной для отрасли еще с прошлого века. Большие сдвиги в процессах разработки (формализация процессов управления требованиями, разделение бизнес- и системного анализа при разработке ПО, итеративность "гибких" методологий) и формализация ролей аналитика и руководителя проекта во многом сократили разрыв между разработчиками, конечными пользователями и заказчиками ПО.

Эксперты в рассматриваемом исследовании отмечают актуальность способов мотивации и вовлечения специалистов-разработчиков в бизнес-вопросы заказчиков и потенциальных пользователей в условиях полностью удаленной работы. Перевод в онлайн практически всех рабочих отношений и коммуникаций нарушил привычное течение производственных процессов на всех этапах реализации IT-проектов. Гипотеза 1 в данном исследовании подразумевает необходимость дополнительных усилий, инвестиций, внимания к таким вопросам на уровне всей компании и отдельных ее проектных команд.

Исследование показало, что выявленный арсенал методов повышения мотивации оказался довольно скудным и не потребовал от компаний, в которых работают эксперты, значительных усилий и больших издержек. К числу таких мотивирующих методов относятся:

1) использование самого факта удаленной работы как мотивационного для программистов и других категорий разработчиков ПО;

2) привлечение внимания к различным дистанционным обучающим курсам и программам (при наличии в компании централизованной функции обучения — как, например, корпоративный университет).

По результатам анализа эксперты отмечают безусловный рост числа и глубины горизонтальных связей, увеличение важности каналов электронного взаимодействия и коммуникаций — новые инструментальные средства, новые группы и чаты мессенджеров, новые вопросы для регулярного обсуждения. Часть экспертов указывает на то, что специалисты

в командах ожидают в больших объемах поддержку от своих компаний, учитывая нарастающий экономический кризис 2020—2021 гг. Подобные ожидания ранее уже были выявлены в исследовании 2020 г. [1].

Важно отметить, что значительным фактором мотивации и демотивации являются внутренние условия для работы специалистов-разработчиков в домашних условиях. Этот фактор включает в себя целый набор обстоятельств личной жизни сотрудников — разработчиков ПО. Эксперты выделили наиболее важные бытовые факторы, благоприятно влияющие на возможности успешной удаленной работы:

- эргономичность рабочего места (стол, кресло, техника, освещенность);
- возможность покоя и концентрации на рабочих вопросах (т. е. ограждение от семьи как фактора стресса, отвлечений и лишнего шума).

С учетом этих факторов вопросы хорошего доступа в Интернет и инфраструктурные вопросы к 2021 г. были успешно решены во всех компаниях, которые представляют эксперты в рассматриваемом исследовании.

Вместе с тем исследование и полученные результаты подтвердили, что в России лишь в незначительном количестве команд и технологических компаний руководители прилагают дополнительные усилия и несут значимые затраты на обеспечение краткосрочной и долгосрочной мотивации разработчиков. Менее 15 % экспертов, принявших участие в исследовании, отметили практическое использование хотя бы одного представленного далее типа соответствующих мероприятий (в порядке убывания распространенности):

1) внедрение практики онлайн командообразования и онлайн корпоративных мероприятий;

2) централизованное внедрение практики повышения уровня знаний инженеров о работе в удаленном формате, включая советы, инструкции и обмен мнениями об актуальных вопросах и способах их разрешения;

3) дополнение усилий команд разработчиков ПО разовыми или регулярными усилиями специалистов вне штата (аутсорсинг, фриланс) в критических точках проектов разработки ПО.

Развитие аналогичных процессов в мировых лидерах IT-отрасли складывается иначе. Компании Google [13], Microsoft [14], Twitter [15] не просто перешли на полностью удаленный формат как стандарт производственной организации, но и прикладывают заметные усилия к повышению эффективности такого формата. Реализуются формы поддержки сотрудников компаний от дополнительных стимулирующих выплат сотрудникам (Fujitsu, Uber) до централизованных политик улучшения мотивации, психологического здоровья и безопасности сотрудников (Google, Facebook, Spotify).

Исследование и анализ его результатов также показали, что в части команд выросло управленческое давление на персонал нижнего звена команд разработчиков ПО, носящее явно выраженную директивную форму. В первую очередь это связано с неблагоприятным экономическим положением таких

компаний. На такие факты указало около 20 % экспертов, принявших участие в исследовании. Часть компаний использует перевод на полностью удаленную работу как повод для аттестации и увольнения наиболее слабых специалистов и сокращения фонда оплаты труда.

Гипотеза 1 в исследовании предполагала не только снижение степени вовлечения программистов в бизнес-вопросы заказчиков и пользователей ПО в кризисных условиях 2020 г., но и возобновление руководством компаний поиска оптимальных подходов к решению задач эффективной удаленной разработки ПО в 2021 г. Однако исследование показало, что в России компании по-прежнему не уделяют должного внимания вопросам долговременного вовлечения программистов в бизнес-вопросы, которые возникают у заказчиков и будущих пользователей разрабатываемого ПО. Лишь в нескольких компаниях были выявлены отмеченные далее организационные мероприятия (в порядке убывания распространенности):

1) дополнительные коммуникационные усилия руководителей проектов как, например, разъяснения бизнес-потребностей заказчиков и пользователей ПО, сопряжение командных результатов с персональными карьерными траекториями, создание дополнительных каналов коммуникаций в проектах;

2) приглашение вернуться в офис вопреки опасностям существующей угрозы пандемии.

Таким образом, исследование указывает на недостаточные усилия, направленные на повышение мотивации и вовлеченности персонала IT-проектов в разрешении смежных для них вопросов после перехода на удаленную разработку ПО. Этот факт означает опровержение положений первой гипотезы исследования. Более того, ослабление заинтересованности персонала, разрабатывающего ПО, в решении реальных бизнес-вопросов заказчиков и пользователей, а не просто в завершение проекта и/или написания программного кода. Это обстоятельство наносит долговременный ущерб отдельным IT-компаниям и снижает в целом конкурентные позиции отечественной отрасли информационных технологий.

Обеспечение условий для профессионального и личного роста исполнителей нижнего звена в командах разработчиков

Профессиональный рост является одним из главных личных мотивационных стимулов для сотрудников IT-отрасли [16]. Переход на удаленную разработку ПО в рамках больших проектов нарушил привычные формы передачи опыта. В гипотезе 2 исследования было принято предположение, что отдельные команды разработчиков, компании, выполняющие проекты, и сами разработчики в новых условиях, обусловленных COVID-экономикой, прилагают дополнительные усилия к созданию благоприятных условий для профессионального роста младшего персонала разработчиков и роста лидерских и управленческих компетенций опытных разработчиков, стимулирующих их переход на более высокие управленческие должности.

Профессиональный рост представителей младшего персонала является долгосрочным и плохо детерминированным процессом [17], который включает в себя:

- получение и закрепление знаний в области IT как инженерных и бизнес-навыков;
- более широкое освоение автоматизируемой предметной области;
- получение детальных представлений о программных продуктах и решениях, создаваемых в конкретной IT-компании.

Такой профессиональный рост необходим самому разработчику, так как именно он определяет его карьерный рост в компании на перспективу 1...2 года с момента найма. Не менее важен он для всего проекта, так как из "обузы" для остальных членов команды исполнитель нижнего звена перерастает в ее полноценного участника.

Практически все эксперты отметили снижение возможностей для профессионального роста начинающих сотрудников IT-компаний с 2020 г. В значительном числе команд и компаний не предпринимаются специальные меры и действия для решения данной задачи. Среди прошедших апробацию способов преодоления отмеченной негативной тенденции (в порядке убывания степени его используемости на практике) следует отметить следующие:

1) наставничество в формате онлайн — от практик "киберспортивных" команд, проводящих весь день в наушниках, до формального перевода всех обучающихся и мотивирующих мероприятий в виртуальную среду, включая замену семинаров на вебинары, появление виртуальных кофе-брейков и т. п.;

2) обеспечение избыточных коммуникаций, дополнительных каналов взаимодействия, специальных слотов времени для обсуждения задач для начинающих специалистов;

3) частичное возвращение в офис вопреки опасностям пандемии.

Некоторые эксперты отметили, что именно социальная изоляция, а не необходимость виртуализации процессов, является ключевым фактором для IT-команд в аспекте обеспечения комфортных условий для профессионального роста сотрудников.

Профессиональный рост опытных сотрудников, связанный с развитием лидерских и управленческих компетенций и сопровождающий их переход на роль лидера команды, в большей степени зависит от мотивации самого сотрудника. Однако конкурентная сила проектных команд и IT-компаний в большой степени зависит именно от такого типа сотрудников [18]. Этот факт означает, что создание благоприятных условий для реализации амбиций и потенциала опытных сотрудников — это конкурентная возможность в развитии компании в IT-отрасли.

Практически все эксперты отметили снижение возможностей в COVID-экономике для обеспечения благоприятных условий роста лидерских и управленческих компетенций у опытных сотрудников, способствующих их карьерному росту. Вместе с тем исследование подтвердило, что большинство российских компаний не предпринимает никаких до-

полнительных усилий для решения этой задачи. Среди причин такого положения дел прежде всего следует отметить недостаток времени и усилий на адаптацию данного сложного процесса к новым условиям организации командной работы. В ходе анализа результатов исследования выявлены следующие мероприятия, направленные на улучшение условий для роста лидерских и управленческих компетенций у опытных сотрудников:

- расширение коммуникаций, включая более строгую формализацию процессов обмена технологическим и управленческим опытом между командами;
- использование дополнительного дистанционного образования в рамках систематизированных курсов для наращивания компетенций опытных сотрудников (например, в рамках корпоративного университета);
- более строгая индивидуализация при решении отмеченной задачи (подбор задач, ролей, проектов) и выстраивание на этой основе персонализированных карьерных траекторий.

Таким образом, положения гипотезы 2 частично подтверждаются. А именно — российские IT-компании понимают необходимость и уже приступили к адаптации процессов создания благоприятных условий для профессионального роста сотрудников в условиях полностью удаленной разработки. Следует отметить, что необходимо больше времени для оценки эффективности усилий в этом направлении.

Перспективы развития форм организации удаленной разработки ПО как нового отраслевого стандарта

Разработка ПО в преимущественно дистанционном режиме де-факто становится новым стандартом в отрасли. Однако и после завершения пандемии для части IT-компаний именно такая организация производственного процесса останется доминирующей. Как следствие, серьезно меняются модели командной работы, условия найма и увольнения.

В проведенном исследовании эксперты выделяют очевидные, по их мнению, достоинства данного подхода, которые останутся значимыми и после завершения пандемии:

- экономия средств и времени при работе из дома;
- возможность сделать рабочий график более гибким и удобным.

Подавляющее большинство экспертов разделяют мнение о справедливости положений гипотезы 3, считая, что в их командах уже определены эффективные механизмы работы в условиях удаленной разработки ПО. Однако вместе с тем эксперты понимают необходимость анализа текущего состояния такой организации разработки и связывают ее перспективы с преодолением определенных проблемных вопросов. В исследовании были выявлены следующие факторы дистанционной разработки, которые можно отнести к числу проблемных:

- зависимость производительности разработчика от обеспечения его рабочих условий (семья —

фактор стресса и фактор поддержки, возможность комфортной работы из дома — шум, свет, рабочее место и т. п.);

- снижение социальной активности работников явление вынужденное как на период пандемии, так и в будущем, после пандемии, при сохранении числа удаленных рабочих мест;

- для каждого сотрудника и членов его семьи — смещение рабочего времени и пространства с личными.

Отмеченные факторы не являются какими-то особенными для IT-отрасли, однако именно технологичные IT-компании, как правило, обладают достаточными ресурсами и опытом в организации распределенных команд разработки [19], позволяющими сгладить или даже разрешить выявленные затруднения. Эксперты в рамках рассматриваемого исследования определили перспективы развития организации полностью удаленной разработки ПО как нового отраслевого стандарта. Для удобства изложения в данной статье они сгруппированы по типам.

Очевидно, что данная форма организации производства уже оказывает значительное влияние на развитие технологий. С одной стороны, IT-компании предъявляют максимально высокие требования к технологическим решениям любых поставщиков. С другой стороны, они сами способны создавать новые технологии, направленные на решение актуальных задач отрасли. Эксперты в исследовании выделили следующие технологические перспективы:

- развитие технологий дистанционных коммуникаций;
- развитие технологий дистанционного обучения, в том числе специализированного;
- кастомизация самими командами технических средств поддержки разработки ПО на различных этапах его жизненного цикла.

Число инструментальных средств дистанционных коммуникаций продолжает увеличиваться, растут требования и ожидания потребителей, усложняются как сами инструментальные средства, так и средства их интеграции с другими информационными системами. Аналогичные процессы наблюдаются в области технологий дистанционного обучения. Еще быстрее в 2020—2021 гг. стали развиваться средства поддержки разработки ПО (SDLM — *Software Development Lifecycle Management*). Есть основания предполагать, что удаленная разработка как формат работы уже в этом году найдет отражение в автоматизации производственных процессов IT-компаний.

Перспективы изменений в организации и методологии проектной и командной работы вызвали высокий уровень интереса при обсуждении с экспертами. Далее приведен сводный список наиболее общих из таких перспектив:

- усиление формализации коммуникационных процессов в IT-компаниях на различных уровнях, включая проектный, корпоративный;
- усиление формализации процессов взаимодействия с бизнес-заказчиками (от демо новых функций

до разбора дефектов), в том числе для обеспечения их прозрачности и предсказуемости;

- повышение роли и значимости социализирующих мероприятий в командах и коллективах — совместные увлечения, корпоративная культура и др.;

- необходимость повышать навыки руководителей компаний в управлении распределенными и дистанционными командами, включая постановку задач, мотивацию, контроль качества продуктов, технологических процессов и т. д.;

- временное упрощение набора бюрократических процедур и возможность их пересмотра и на будущее — замены в компаниях, выбравших информационную безопасность и бюрократию в ущерб удобству сотрудников и гибкости процессов (банки, страховые компании и т. п.);

- необходимость синхронизации графиков дистанционно работающих сотрудников (в командах, подразделениях и т. п.);

- необходимость еще большего погружения менеджеров в детали поставленных задач, особенно при работе с новыми членами их команд.

Разнонаправленное влияние нового организационного формата на строгость исполнения процессов следует рассматривать как универсальную оптимизацию. В технологических отраслях такие оптимизации происходят регулярно [20]. Еще интереснее отметить рост ожиданий инженеров и руководителей, связанный с необходимым ростом управленческих компетенций менеджеров в новых условиях. Очевидно, что все участники понимают временную неоптимальность процессов и ожидают от своих компаний централизованного обновления в моделях управления производственными процессами и отдельными работниками.

Вопросом на перспективу остается необходимость определения оптимальной эффективности коммуникаций в проектной команде (частота, продолжительность, каналы связи и т. д.) в условиях полностью дистанционной работы. Очевидно, что менеджеры, нашедшие правильные параметры для своих проектных команд, раньше других получают тактическое преимущество в развитии своих продуктов и сервисов.

Заключение

Проведенное исследование позволило прийти к ряду общих выводов, актуальных в среднесрочной перспективе развития отечественной IT-отрасли. Во-первых, несмотря на усилия отдельных руководителей и лидеров команд, российские технологические компании уделяют недостаточно внимания, затрачивают минимум усилий и инвестиций для более глубокой адаптации команд к дистанционной работе. При сохранении такого положения дел в организации производства ПО, российская IT-отрасль будет уступать в конкурентной борьбе (наем и удержание талантов, эффективность разработки ПО, развитие команд и сотрудников и т. д.).

Несмотря на это результаты исследования подтверждают, что в российских компаниях происходит поиск подходов к адаптации сотрудников для работы в условиях удаленной разработки. Недостаток ресурсов и возможностей на уровне команд позволяет оценивать попытки изменить положение дел как минимально достаточные для сохранения текущих показателей эффективности работы разработчиков. Понимание необходимости последовательной работы с мотивацией, вовлечением и профессиональным развитием сотрудников в условиях COVID-экономики в российских компаниях не является доминирующим мотивом для корпоративных инвестиций в человеческий капитал. При этом проблемы обмена опытом и профессионального роста разработчиков ПО в новых условиях очевидны экспертам. Однако они требуют большего времени для поиска оптимальных решений и успешного преодоления сложившейся ситуации.

Следует сделать аналогичный вывод о необходимости решения вопросов социализации разработчиков в условиях полностью удаленной разработки. Проблема очевидна экспертам, на ее решение в различных формах направлены значительные усилия менеджмента.

Удаленная разработка и необходимость поддержания ее эффективности оказывают значимое давление на процессные модели в IT-командах. С одной стороны, сложные бюрократические процессы временно (или навсегда) упрощаются, с другой стороны, усиливаются подходы к более строгому описанию коммуникаций как внутри команды, так и с другими командами, заказчиками и пользователями ПО.

Представленное исследование частично подтвердило сделанные ранее выводы исследования [1] — переход на удаленную разработку ПО поддерживается программистами и менеджерами, он оказывает позитивное мотивационное воздействие на работу проектных команд, рассматривается большинством участников процесса разработки как новый организационный стандарт после завершения пандемии.

Список литературы

1. Пашенко Д. С., Комаров Н. М. Ускорение виртуализации рабочих процессов в отраслях "новой экономики" под воздействием пандемии COVID-19 как один из экономических трендов в 2020 году // Вопросы экономики. — 2021. — № 9. — В печати.
2. Цифровые технологии и кибербезопасность в контексте распространения COVID-19. Дайджест Счетной Палаты РФ. URL: <https://ach.gov.ru/upload/pdf/Covid-19-digital.pdf>
3. Сафонов А. Ю. Управление персоналом в период пандемии // Московский экономический журнал. — 2020. — № 5. — С. 554—558.
4. Ralph P., Baltés S., Adisaputri G. et al. Pandemic programming // Empirical Software Engineering. — 2020. — Vol. 25, Iss. 6. — P. 4927—4961. DOI: 10.1007/s10664-020-09875-y

5. Core R. JRebel. The COVID-19 Developer Impact Report. URL: <https://www.jrebel.com/resources/covid-19-developer-impact-report>

6. Игнатъева Я. А., Андрищенко О. В. Система мотивации в IT-компаниях: зарубежный и отечественный опыт // Парадигма. — 2019. — № 3. — С. 37—43.

7. Устюгова Е. Лучшие практики подбора и мотивации IT-специалистов. Отраслевое исследование KPMG. URL: <https://mustread.kpmg.ru/articles/motivatsiya-it-i-digital-spetsialistov/>

8. Вторушин Н. А., Потапов А. М. Мотивация и стимулирование трудовой деятельности: настоящее и будущее // Вестник науки Сибири. — 2015. — № 4 (19). — С. 65—70.

9. Дюпина Е. Как выстроить систему наставничества и развития сотрудников. Опыт "Акелон" // ECM Journal. 30.05.2020. URL: <https://ecm-journal.ru/docs/Kak-vystroit-sistemunastavnichestva-i-razvitiya-sotrudnikov-Opyt-Akelon.aspx>

10. Крицкая М. Как правильно ставить задачи, чтобы сотрудники их выполняли // Журнал Контур. 16.11.2018. URL: <https://kontur.ru/articles/5291>

11. Пашенко Д. С. Как инженеры софтверных компаний воспринимают производственные изменения // Мир новой экономики. — 2015. — № 1. — С. 74—82.

12. Пашенко Д. С. Географически распределенные команды: естественные и организационные особенности проектов разработки программного обеспечения // Программная инженерия. — 2017. — Т. 8, № 2. — С. 88—95.

13. Перельгин Е. Политика удаленной работы Google содержит 6 отличных советов, которые вы должны применить уже сегодня // Портал VC. Раздел Карьера. 28.10.2020. URL: <https://vc.ru/hr/171424-politika-udalenoj-raboty-google-soderzhit-6-otlichnyh-sovetov-kotorye-vy-dolzny-primenit-uzhe-segodnya>

14. Warren T. Microsoft is letting more employees work from home permanently // The Verge. 09.10.2020. URL: <https://www.theverge.com/2020/10/9/21508964/microsoft-remote-work-from-home-covid-19-coronavirus>

15. Kantrowitz A. Twitter Will Allow Employees To Work At Home Forever // BuzzFeed.News Trends. 12.05.2020. URL: <https://www.buzzfeednews.com/article/alexkantrowitz/twitter-will-allow-employees-to-work-at-home-forever>

16. Ерофеев Е. В. Разработка инструментария для мотивации работников организации информационных технологий // Интернет-журнал Науковедение. — 2014. — № 1 (20). — URL: https://www.elibrary.ru/download/elibrary_21541335_97372567.pdf

17. Черепанов С. Как карта развития помогает справляться со стихийными повышениями и неосознанной некомпетентностью // Портал VC. Раздел Карьера. 25.02.2021. URL: <https://vc.ru/hr/212949-kak-karta-razvitiya-pomogaet-spravlyatsya-so-stihiynymi-povysheniyami-i-neosoznannoy-nekompetentnostyu>

18. Bachuk A. The path to technical leadership: how to go from developer to team leader // 27.12.2018, URL: <https://www.freecodecamp.org/news/the-path-to-technical-leadership-how-to-go-from-developer-to-team-leader-8c544f15a431/>

19. Calvi T. T. F., Moriya H. E. N. Software Project Management in Distributed Software Development Context // In Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS-2013), 2013. — P. 216—222. URL: <https://www.scitepress.org/Papers/2013/44424/44424.pdf>

20. Pashchenko D. Digitalization in Software Engineering and IT Business // International Journal of Software Science and Computational Intelligence. — 2020. — Vol. 12, Iss. 2. — Article 1. DOI: 10.4018/IJSSCI.2020040101.

Russian Experience in Organizing Fully Remote Software Development: an Industry Study of 2021

D. S. Pashchenko, denpas@rambler.ru, Moscow, 125368, Russian Federation

Corresponding author:

Pashchenko Denis S. Moscow, 125368, Russian Federation

E-mail: denpas@rambler.ru

Received on April 29, 2021

Accepted on June 07, 2021

The pandemic and the COVID-economy continue to have a significant impact on the development of industries of the "new economy". We might see it in the acceleration of digitalization, virtualization of work processes and the predominance of communications in electronic channels. The transition to fully remote development and support of software under the influence of the COVID factor not just changed the IT industry, but established a new standard of organization of the software production. This article presents the results of study that covered various aspects of fully remote software development in Russian companies, including VTB, Yandex, Sbertech, and others. More than 20 experts representing their development teams have identified the main approaches to ensuring the motivation and involvement of engineers and to ensuring favorable conditions for their professional growth. The study also identifies the prospects for the evolution of the fully remote software development as a new industry standard. The results of the study are accompanied by the brief conclusions, the main idea of which is the need to quickly understand a new trend in software development and the usage of this factor in the competitive struggle (hiring specialists, organizing teams, organizational development).

Keywords: remote work, information technology, software development, COVID-economy, software technology

For citation:

Pashchenko D. S. Russian Experience in Organizing Fully Remote Software Development: an Industry Study of 2021, *Programmnaya Ingereria*, 2021, vol. 12, no. 6, pp. 311–318.

DOI: 10.17587/prin.12.311-318

References

1. Pashchenko D. S., Komarov N. M. Acceleration of virtualization of work processes in the branches of the "new economy" under the influence of the COVID-19 pandemic, as one of the economic trends in 2020, *Voprosy ekonomiki*, 2021, no. 9, in print (in Russian).
2. Digital technologies and cybersecurity in the context of the spread of COVID-19. Djest of the Accounts Chamber of the Russian Federation, available at: <https://ach.gov.ru/upload/pdf/Covid-19-digital.pdf> (in Russian).
3. Safonov A. Yu. Personnel management during a pandemic, *Moskovskij ekonomicheskij zhurnal*, 2020, no. 5, pp. 554–558 (in Russian).
4. Ralph P., Baltes S., Adisaputri G. et al. Pandemic programming, *Empirical Software Engineering*, 2020, vol. 25, iss. 6, pp. 4927–4961, DOI: 10.1007/s10664-020-09875-y
5. Cope R. JRebel. The COVID-19 Developer Impact Report, available at: <https://www.jrebel.com/resources/covid-19-developer-impact-report>
6. Ignatieva Ya. A., Andryushchenko O. V. Motivation system in IT companies: foreign and domestic experience, *Paradigma*, 2019, no. 3, pp. 37–43 (in Russian).
7. Ustyugova E. Best practices for recruiting and motivating IT specialists. KPMG Industry Research, available at: <https://mustread.kpmg.ru/articles/motivatsiya-it-i-digital-spetsialistov/> (in Russian).
8. Vtorushin N. A., Potapov A. M. Motivation and stimulation of labor activity: present and future, *Vestnik nauki Sibiri*, 2015, no. 4 (19), pp. 65–70 (in Russian).
9. Dupina E. How to build a system of mentoring and development of employees. The Akelon Experience, *ECM Journal*, 30.05.2020, available at: <https://ecm-journal.ru/docs/Kak-vystroit-sistemu-nastavnichestva-i-razvitiya-sotrudnikov-Opyt-Akelon.aspx> (in Russian).
10. Kritskaya M. How to set tasks correctly so that employees perform them, *Kontur Journal*, 16.11.2018, available at: <https://kontur.ru/articles/5291> (in Russian).
11. Pashchenko D. S. How software engineers perceive production changes, *Mir novoj ekonomiki*, 2015, no. 1, pp. 74–82 (in Russian).
12. Pashchenko D. S. Geographically Distributed Teams: Natural and Organizational Features of Software Development Projects, *Programmnaya Ingereria*, 2017, vol. 8, no. 2, pp. 88–95 (in Russian).
13. Perelygin E. Google's Remote Work Policy contains 6 great tips that you should apply today, *VC Portal. Career section*, 28.10.2020, available at: <https://vc.ru/hr/171424-politika-udalenny-raboty-google-soderzhit-6-otlichnyh-sovetov-kotorye-vy-dolzny-primenit-uzhe-segodnya> (in Russian).
14. Warren T. Microsoft is letting more employees work from home permanently, *The Verge*, 09.10.2020, available at: <https://www.theverge.com/2020/10/9/21508964/microsoft-remote-work-from-home-covid-19-coronavirus>
15. Kantrowitz A. Twitter Will Allow Employees To Work At Home Forever, *BuzzFeed.News Trends*, 12.05.2020, available at: <https://www.buzzfeednews.com/article/alexkantrowitz/twitter-will-allow-employees-to-work-at-home-forever>
16. Erofeev E. V. Development of tools for motivating employees of information technology organizations, *Internet-zhurnal "Naukovedenie"*, 2014, no. 1 (20), available at: https://www.elibrary.ru/download/elibrary_21541335_97372567.pdf (in Russian).
17. Cherepanov S. How the development map helps to cope with spontaneous promotions and unconscious incompetence, *VC Portal. Career section*, 25.02.2021, available at: <https://vc.ru/hr/212949-kak-karta-razvitiya-pomogaet-spravlyatsya-so-stihiynymi-povysheniyami-i-neosoznannoy-nekompetentnostyu> (in Russian).
18. Bachuk A. The path to technical leadership: how to go from developer to team leader, *freeCodeCamp. Leadership*, 27.12.2018, available at: <https://www.freecodecamp.org/news/the-path-to-technical-leadership-how-to-go-from-developer-to-team-leader-8c544f15a431/>
19. Calvi T. T. F., Moriya H. E. H. Software Project Management in Distributed Software Development Context, *In Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS-2013)*, 2013, pp. 216–222, available at: <https://www.scitepress.org/Papers/2013/44424/44424.pdf>
20. Pashchenko D. Digitalization in Software Engineering and IT Business, *International Journal of Software Science and Computational Intelligence*, 2020, vol. 12, iss. 2, article 1. DOI: 10.4018/IJSSCI.2020040101.

А. С. Акопов, д-р техн. наук, проф., гл. науч. сотр.¹, проф.², akopovas@umail.ru,

Н. К. Хачатрян¹, канд. физ.-мат. наук, зам. директора, nerses@cemi.rssi.ru

¹ЦЭМИ РАН, Москва

²Национальный исследовательский университет "Высшая Школа Экономики", Москва

Имитационная модель нефтеперерабатывающего предприятия со сложной схемой взаимодействия оборудования*

Представлена разработанная авторами имитационная модель нефтеперерабатывающего предприятия (НПЗ). Особенностью этой модели является применение методов агентного и дискретно-событийного моделирования. Другой особенностью является использование эвристической оптимизации для поиска наилучших значений ключевых производственных и ресурсных характеристик нефтеперерабатывающих установок, взаимодействующих в рамках сложной производственной схемы построения НПЗ, обеспечивающей глубокий уровень нефтепереработки. При этом целевой функцией предлагаемой модели является прибыль предприятия, которая учитывает как выручку от реализации нефтепродуктов, так и эксплуатационные затраты. Предложенная модель реализована в системе сопровождения имитационного моделирования AnyLogic. Она позволяет, в частности, максимизировать прибыль НПЗ при различных ограничениях, в том числе на объемы поставок нефтепродуктов, необходимые для выполнения контрактных обязательств в заданные моменты времени.

Ключевые слова: имитационное моделирование производственных процессов, нефтепереработка, дискретно-событийное моделирование, агентное моделирование, оптимальное производственное планирование, AnyLogic

Введение

В настоящее время наблюдается существенное усложнение бизнес-процессов в нефтеперерабатывающих предприятиях (НПЗ), обусловленное в основном необходимостью перехода к рациональной, ресурсосберегающей и энергоэффективной производственной стратегии. При этом крупные НПЗ для поддержки операционной деятельности внедряют современные системы оптимального производственного планирования, например, Aspen Plus [1], Symphonite (RPMS) [2] и др. Подобные системы основаны на использовании методов математического программирования, в частности, известных алгоритмов оптимального составления расписаний загрузки производственного оборудования в технологической цепочке: от термического крекинга (высокотемпературной переработки нефти в целях получения продуктов меньшей молекулярной массы) до компаундирования бензинов (смешивания бензиновых фракций) [3–7]. При этом существующие информационные системы операционного управления НПЗ используют в основном методы линейного

программирования для оптимальной балансировки загрузки производственных установок при имеющихся ограничениях на общий объем поставки сырья, требуемые сроки и объемы поставок конечной продукции [8]. Вместе с тем динамика поставок сырья как и спрос на конечную продукцию носит более сложный нелинейный характер с наличием множественных временных разрывов во входных и исходящих потоках. Такие разрывы обусловлены вероятностным простоем используемого оборудования, являющимся следствием аварийных ситуаций, либо сбоями в логистической системе. Подобные трудности приводят к отсутствию возможности корректного применения линейных оптимизационных моделей без потерь в качестве получаемых решений. Поэтому необходима разработка имитационных агент-ориентированных моделей производственных процессов, учитывающих как стохастический характер входных и выходных материальных потоков, так и скорость процессов переработки сырья в технологической цепочке. Такая переработка зависит от множества влияющих факторов, например, среднее время производственного цикла, количество задействованных ресурсов и др. В отличие от жесткой линейной системы производственного планирования, имитационная модель НПЗ может быть использована как инструмент поиска оптимальных технологических

* Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (РФФИ) в рамках научного проекта № 19-010-00958.

Сравнение предлагаемого подхода с известными аналогами

Ключевые особенности системы	Имитационная модель НПЗ (AnyLogic)	Aspen Plus	CHEMCAD	Hysys и Hysim
Поддержка возможности оптимизации производственных и ресурсных характеристик НПЗ на основе имитационного моделирования	Да (возможность выбора целевой функции и ограничений)	Частично	Частично	Нет
Поддержка дискретно-событийного и агентного моделирования	Да	Нет	Нет	Нет
Моделирование сложных физико-химических процессов	Нет	Да	Да	Да
Расчет технологических схем, построенных из отдельных элементов	Частично	Да	Да	Да
Тип системы (возможность модификаций)	Открытая	Частично открытая	Частично открытая	Частично открытая
Стоимость системы	Низкая	Высокая	Высокая	Высокая

решений в условиях существенной волатильности и разрывности в динамике поставок сырья, вероятностного отключения отдельного оборудования из производственной цепочки вследствие аварий и др.

Особенностью работы современных НПЗ [5, 6] является возможность реконфигурирования так называемой схемы построения НПЗ, т. е. конфигурации взаимодействия нефтеперерабатывающих установок в зависимости от производственно-сбытовой стратегии предприятия. Другой важной особенностью является возможность перераспределения потоков переработанного сырья между текущим сбытом и производственными установками в целях максимизации валовой прибыли НПЗ.

Отметим, что в настоящее время методы математического и имитационного моделирования успешно применяются для рационального управления сложными производственными и экономическими процессами. Применение таких методов необходимо, например, для прогнозирования качества [9] и оптимизации скорости смешивания разнооктановых бензинов [10], оптимального суточного планирования производства нефтепродуктов [11], оптимизации энергопотребления производственных установок [12] и др.

При этом применение методов агентного и дискретно-событийного моделирования позволяет учесть специфику исследуемых производственных процессов. Среди примеров применения систем имитационного моделирования для управления характеристиками перерабатывающего предприятия можно выделить работы [13–15]. В этих работах, в частности, отмечены преимущества агентного подхода по сравнению с традиционными балансовыми моделями. Методы агент-ориентированного моделирования (АОМ) позволяют исследовать поведение сложных децентрализованных систем, в которых реализуется многочастичное взаимодействие и трансформация (т. е. изменение состояния) агентов в зависимости от заданных условий и ограничений [16–18]. Подобный

подход также позволяет моделировать сложные физико-химические процессы, в которых наблюдается трансформация веществ под воздействием внешних факторов [17], что особенно важно для изучения эффективности процессов нефтепереработки. В табл. 1 в укрупненном виде представлено сравнение предлагаемого в статье подхода с известными аналогами.

Из данных табл. 1 следует, что предлагаемый подход (имитационная модель НПЗ в AnyLogic) имеет преимущество по сравнению с известными аналогами (например, Aspen Plus, CHEMCAD, Hysys и Hysim) в части возможности оптимизации производственных и ресурсных характеристик НПЗ на основе имитационного моделирования, поддержки методов дискретно-событийного и агентного моделирования, открытости платформы для внесения модификаций и более низкой стоимости внедрения. Основным недостатком при этом является отсутствие готовых алгоритмов расчета технологических процессов и поддержки моделирования сложных физико-химических процессов.

В статье представлена разработанная авторами агент-ориентированная имитационная модель нефтеперерабатывающего предприятия со сложной схемой взаимодействия оборудования с реализацией в системе имитационного моделирования AnyLogic. Рассматриваемое предприятие обеспечивает глубокий уровень переработки и производство широкого ассортимента нефтепродуктов. Подобная модель может быть использована в качестве системы поддержки принятия оперативных решений, в том числе для оптимального суточного производственного планирования, на нефтеперерабатывающих предприятиях.

1. Имитационная модель НПЗ

На рис. 1 представлена укрупненная схема предлагаемой имитационной модели НПЗ, построенной по сложной схеме. Ранее подобная схема использовалась в исследованиях, результаты которых изложены

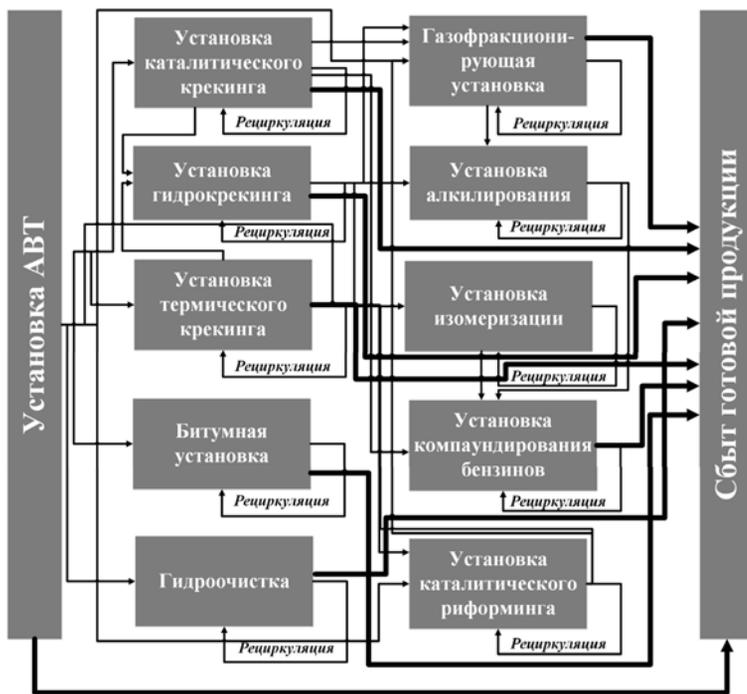


Рис. 1. Укрупненная схема имитационной модели НПЗ

в работе [4]. При этом особенности функционирования соответствующих производственных установок описаны в работах [5, 6].

Ключевым элементом разработанной имитационной модели НПЗ (рис. 1) является установка АВТ (атмосферно-вакуумная трубчатка), обеспечивающая первичную переработку нефти и ее разделение на узкие фракции для последующей гидроочистки, крекинга и риформинга. Последняя из перечисленных технологий необходима для получения высокооктановых бензинов и ароматических углеводородов. Как правило, выходной материальный поток каждой производственной установки (например, установки АВТ) является входным потоком для других установок (например, установки гидроочистки, битумной установки и т. д.). При этом учитываются процессы рециркуляции остаточного продукта, влияющие на динамику выходных потоков. В подобной схеме можно выделить два класса продуктов — первичной и вторичной нефтепереработки соответственно. Первичная переработка осуществляется установкой АВТ с последующей гидроочисткой и крекингом. Вторичная переработка при этом обеспечивается преимущественно установками риформинга и компаундирования бензинов.

В предложенной имитационной модели НПЗ агентами являются емкостные объекты (например, баррели), содержащие различные химические вещества (простые и высокомолекулярные углеводородные соединения), среди которых можно выделить низко- и высокооктановые бензины, авиационный керосин, дизельное топливо, битум, мазут, гудрон, полипропилен, синтетические масла, ароматизаторы и др. Таким образом, получаемые в результате многочастичного взаимодействия агенты-нефте-

продукты попадают в систему сбыта готовой продукции (рис. 1). Подобный подход, основанный на взаимодействии агентов-частиц, используется при моделировании поведения крупномасштабных децентрализованных систем [17]. Помимо агентов-нефтепродуктов в составе модели используются агенты — сотрудники НПЗ, задействованные в обслуживании различного нефтеперерабатывающего оборудования.

Приведем краткое формальное описание основных характеристик предложенной модели.

Пусть

- T — набор временных моментов (в мин); $|T|$ — общее число временных моментов; $t_0 \in T$, $t_{|T|} \in T$ — начальные и конечные моменты времени, $k = 0, \dots, |T|$ — все моменты времени;

- $I = \{i_1, i_2, \dots, i_{|I|}\}$ — набор индексов агентов-продуктов первичной переработки (например, мазут, битум, керосин и т. д.), где $|I|$ — общее количество продуктов первичной переработки;

- $J = \{j_1, j_2, \dots, j_{|J|}\}$ — набор индексов агентов-продуктов вторичной переработки (например, высокооктановый бензин, нефтяные масла и т. д.), где $|J|$ — общее число продуктов вторичной переработки;

- $\sigma(\bar{o}, \bar{\sigma})$ — среднесуточный объем сырой нефти, подаваемый на вход НПЗ, равномерно распределенный на интервале $[\bar{o}, \bar{\sigma}]$;

- $\{o_i(t_k), \bar{o}_{ij}(t_k)\}$, $t_k \in T$, $i \in I$, $j \in J$ — интенсивность поставок емкостных объемов сырой нефти, подаваемой на вход установки по производству i -го продукта, а также поставок i -го продукта первичной переработки на вход установки по производству j -го продукта вторичной переработки в момент времени t_k (*управляющий параметр модели*);

- $\alpha_i, \bar{\alpha}_j \in [0, 1]$, $i \in I$, $j \in J$ — доля i -го и j -го продуктов, направляемая в систему сбыта для реализации, вместо возможной переработки (*управляющий параметр модели*);

- $\{g_i, \bar{g}_{ij}\}$, $i \in I$, $j \in J$ — количество единиц (емкостных объемов) сырой нефти, необходимых для производства единицы i -го продукта и единиц i -го продукта, необходимых для производства единицы j -го продукта;

- $\{v_i(t_k), \bar{v}_j(t_k)\}$, $t_k \in T$, $i \in I$, $j \in J$ — количество единиц (емкостных объемов) i -го и j -го продуктов, производимых в момент времени t_k ;

- $\{\delta t_i, \bar{\delta t}_{ij}\}$, $i \in I$, $j \in J$ — интервал времени, прошедший от начала запуска производственного процесса на установках по производству i -го и j -го продуктов соответственно;

- $\gamma_i, \bar{\gamma}_j \in \{0, 1\}$, $i \in I$, $j \in J$ — матрицы "включений" нефтеперерабатывающего оборудования в схему построения НПЗ (*управляющий параметр модели*): 0 — оборудование не используется, 1 — установка включена в схему НПЗ;

- $\{t_i^*, \bar{t}_j^*\}$, $i \in I$, $j \in J$ — моменты отгрузки конечной продукции для i -го и j -го продуктов соответственно;

- $\{d_i(t_i^*), \tilde{d}_j(\tilde{t}_j^*)\}, i \in I, j \in J$ — требуемый объем производства, определяемый контрактными обязательствами в моменты времени $t_i^*, \tilde{t}_j^* \in T$;

- P_j — множество продуктов первичной переработки, необходимых для производства единицы j -го продукта вторичной переработки;

- $\{c_i, \tilde{c}_{ij}\}, i \in I, j \in J$ — стоимость сырья, необходимого для производства единицы i -го (стоимость сырой нефти) и j -го продуктов (трансфертная стоимость продуктов первичной переработки);

- $\{w_i(t_k), \tilde{w}_j(t_k)\}, i \in I, j \in J$ — затраты на оплату труда производственного персонала, задействованного в производстве i -го и j -го продуктов соответственно;

- $\{L_i, \tilde{L}_j\}, i \in I, j \in J$ — число производственного персонала, задействованного в производстве i -го и j -го продуктов соответственно (управляющий параметр модели);

- $\{\tau_i(\tau_i^*, \sigma_i), \tilde{\tau}_j(\tilde{\tau}_j^*, \tilde{\sigma}_j)\}, i \in I, j \in J$ — время, необходимое для производства единицы i -го и j -го продуктов, заданное в соответствии с нормальным распределением с математическими ожиданиями $\tau_i^*, \tilde{\tau}_j^*$ (управляющие параметры модели) и стандартными отклонениями $\sigma_i, \tilde{\sigma}_j$ для i -го и j -го продуктов соответственно;

$$\sigma_i = \begin{cases} 1 - \frac{L_i}{L_i^*}, & \text{если } L_i < L_i^*, \\ 0, & \text{если } L_i \geq L_i^*, \end{cases} \quad \tilde{\sigma}_j = \begin{cases} 1 - \frac{\tilde{L}_j}{\tilde{L}_j^*}, & \text{если } \tilde{L}_j < \tilde{L}_j^*, \\ 0, & \text{если } \tilde{L}_j \geq \tilde{L}_j^*, \end{cases}$$

где $L_i^*, \tilde{L}_j^*, i \in I, j \in J$ — нормативное число сотрудников, необходимое для работы на установках по производству i -го и j -го продуктов соответственно;

- $\{p_i, \tilde{p}_j\}, i \in I, j \in J$ — конечная стоимость i -го и j -го продуктов соответственно.

Тогда объемы производства продуктов первичной и вторичной переработки в момент времени $t_k, \tilde{t}_k \in T$:

$$v_i(t_k) = \begin{cases} o_i(t_k), & \text{если } o_i(t_k) \geq g_i \text{ и } t_k - \delta t_i = \tau_i \text{ и } \gamma_i = 1, \\ 0, & \text{если } t_k - \delta t_k \neq \tau_i \text{ или } \gamma_i = 0, \end{cases}$$

$$\tilde{v}_j(t_k) = \begin{cases} \sum_{i=1}^{|J|} \tilde{o}_{ij}(t_k), & \text{если } \tilde{o}_{ij}(t_k) \geq \tilde{g}_{ij} \text{ для всех } i \in P_j \\ \text{и } t_k - \delta \tilde{t}_j = \tilde{\tau}_j \text{ и } \tilde{\gamma}_j = 1, \\ 0, & \text{если } t_k - \delta \tilde{t}_j \neq \tilde{\tau}_j \text{ или } \tilde{\gamma}_j = 0. \end{cases}$$

Выручка от реализации продуктов первичной и вторичной переработки, определяемая в моменты отгрузки конечной продукции:

$$V(t_i^*) = \sum_{i=1}^{|I|} \sum_{t_k=1}^{t_i^*} v_i(t_k) p_i, \quad \tilde{V}(\tilde{t}_j^*) = \sum_{i=1}^{|I|} \sum_{t_k=1}^{\tilde{t}_j^*} \tilde{v}_j(t_k) \tilde{p}_j,$$

$$t_i^*, \tilde{t}_j^* \in T, i \in I, j \in J.$$

Затраты на производство продуктов первичной и вторичной переработки:

$$C(t_i^*) = \sum_{i=1}^{|I|} \sum_{t_k=1}^{t_i^*} (o_i(t_k) c_i + L_i w_i(t_k)),$$

$$\tilde{C}(\tilde{t}_j^*) = \sum_{j=1}^{|J|} \sum_{t_k=1}^{\tilde{t}_j^*} \left(\sum_{i=1}^{|I|} \tilde{o}_{ij}(t_k) \tilde{c}_{ij} + L_j w_j(t_k) \right).$$

Валовая прибыль НПЗ (без учета налогов и сборов):

$$\pi = \sum_{t_i^* \in T} (V(t_i^*) - C(t_i^*)) + \sum_{\tilde{t}_j^* \in T} (\tilde{V}(\tilde{t}_j^*) - \tilde{C}(\tilde{t}_j^*)).$$

Теперь можно сформулировать основную оптимизационную задачу, решение которой обеспечивается имитационным моделированием НПЗ.

Задача. Требуется максимизировать валовую прибыль НПЗ по наборам управляющих параметров $\{\gamma_i, \tilde{\gamma}_j\}, \{\alpha_i, \tilde{\alpha}_j\}, \{o_i, \tilde{o}_{ij}\}, \{L_i, \tilde{L}_j\}, \{\tau_i^*, \tilde{\tau}_j^*\}$:

$$\pi \rightarrow \max_{\{\gamma_i, \tilde{\gamma}_j\}, \{\alpha_i, \tilde{\alpha}_j\}, \{o_i, \tilde{o}_{ij}\}, \{L_i, \tilde{L}_j\}, \{\tau_i, \tilde{\tau}_j\}} \quad (1)$$

при ограничениях

$$\sum_{t_k=1}^{t_i^*} v_i(t_k) = d_i(t_i^*), \quad \sum_{t_k=1}^{\tilde{t}_j^*} \tilde{v}_j(t_k) = \tilde{d}_j(\tilde{t}_j^*), \quad (2)$$

$$\underline{L} \leq \sum_{i=1}^{|I|} L_i + \sum_{j=1}^{|J|} L_j \leq \bar{L}, \quad (3)$$

$$\underline{\tau}_i \leq \tau_i^* \leq \bar{\tau}_i, \quad \underline{\tau}_j \leq \tau_j^* \leq \bar{\tau}_j, \quad 0 \leq \alpha_i, \tilde{\alpha}_j \leq 1, \quad (4)$$

$$t_i^*, \tilde{t}_j^* \in T, \gamma_i, \tilde{\gamma}_j \in \{0, 1\}, i \in I, j \in J. \quad (5)$$

Здесь $\underline{L}, \underline{\tau}_i, \underline{\tau}_j, \{\bar{L}, \bar{\tau}_i, \bar{\tau}_j\}, i \in I, j \in J$ — нижние и верхние граничные значения управляющих параметров модели.

Сложность задачи (1)–(5) обусловлена необходимостью рационального управления распределением потоков переработанного сырья между текущим сбытом и производственными установками $\{\alpha_i, \tilde{\alpha}_j\}, i \in I, j \in J$, среднесуточными интенсивностями загрузки оборудования сырьем $\{o_i, \tilde{o}_{ij}\}, i \in I, j \in J$, учета имеющихся контрактных обязательств, заданных для отдельных моментов времени $t_i^*, \tilde{t}_j^* \in T, i \in I, j \in J$ и выбора наиболее предпочтительной схемы построения НПЗ посредством матриц "включения" перерабатывающих установок $\{\gamma_i, \tilde{\gamma}_j\}, i \in I, j \in J$ и др. При этом время, необходимое для производства каждого продукта первичной и вторичной переработки, ограничено технологическими возможностями каждой установки и зависит от уровня загрузки производственно-промышленного персонала (в частности, увеличивается в условиях дефицита трудовых ресурсов). Некоторые характеристики модели (например, суточный объем поставок нефти на вход НПЗ, время работы установок и др.) также задаются в соответствии с равномерным

и нормальными статистическими распределениями. Поэтому данная задача относится к классу задач стохастической оптимизации большой размерности и решается с использованием генетического алгоритма, агрегированного по целевому функционалу и ограничениям с предложенной имитационной моделью НПЗ, реализованной в AnyLogic.

2. Программная реализация модели

На рис. 2 представлена укрупненная архитектура спроектированной системы управления производственными и ресурсными характеристиками НПЗ, построенной на основе предложенной имитационной модели.

Разработанная система использует предложенную авторами имитационную модель НПЗ со сложной схемой взаимодействия оборудования, реализованную в системе AnyLogic. Таким образом, спроек-

тированная система включает перечисленные далее важные компоненты:

- **имитационную модель НПЗ**, обеспечивающую вычисление значения целевой функции системы (прибыли), суточных объемов производства нефтепродуктов (по видам) и другие характеристики при различных сценарных условиях и ограничениях;
- **генетический оптимизационный алгоритм**, предназначенный для поиска оптимальных значений целевой функции, являющейся результатом имитационного моделирования, и устойчивый по отношению к стохастической динамике поставок сырой нефти на вход НПЗ и вероятностным характеристикам параметров, определяющих скорости производственных циклов перерабатывающих установок; применяемый генетический алгоритм относится к классу стандартных эвристических алгоритмов бинарного кодирования, предназначенных для одноцелевой оптимизации [19], при этом значение целевой

Панель управления моделью

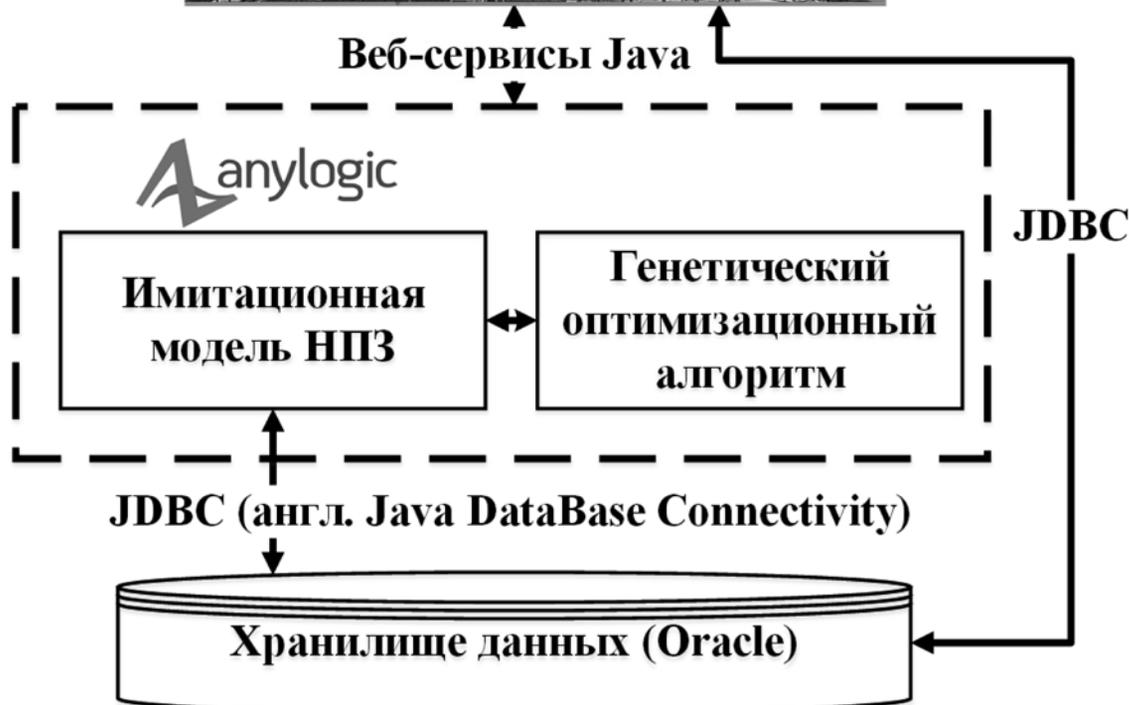


Рис. 2. Укрупненная архитектура информационной системы производственного планирования НПЗ

функции вычисляется в результате имитационного моделирования и используется для формирования новых потенциальных решений посредством операторов селекции, кроссовера и мутации;

- **хранилище данных**, предназначенное для информационного обеспечения имитационной модели исходными данными (технологические параметры производственных установок, данные по объемам поставок сырья в предыдущие периоды времени и др.), а также сохранения результатов имитационного и оптимизационного моделирования;

- **панель управления моделью**, представляющую собой набор веб-интерфейсов, обеспечивающих возможность сценарного изменения значений управляющих параметров модели, запуска расчетов целевых характеристик модели (при заданной схеме построения НПЗ), выполнение оптимизационных экспериментов и др.

На рис. 3. представлен ключевой фрагмент предложенной имитационной модели НПЗ с реализацией в AnyLogic.

Особенностью реализации имитационной модели НПЗ в AnyLogic (рис. 3) является использование специальной программной библиотеки моделирования потоков (*Fluid Library*). Данная библиотека позволяет

моделировать процессы хранения, транспортировки и смешивания жидкостей, объемного вещества и множества дискретных элементов (агентов-продуктов). Подобные жидкостные потоки могут быть трансформированы в агентов и обратно для реализации правил перехода к новым состояниям модели. При этом предоставляется возможность управлять как объемами сырья, так и скоростями их смешивания для каждой взаимодействующей установки.

Для обеспечения имитационной модели актуальными данными, описывающими ключевые характеристики используемого оборудования (например, нормативное время переработки емкостной единицы углеводородного сырья), а также динамику поставок сырья и нефтепродуктов, было спроектировано многомерное информационное хранилище с множественными таблицами фактов и измерений с реализацией в СУБД Oracle. Интеграция имитационной модели с Oracle обеспечивается посредством JDBC API, которая представляет собой систему драйверов и интерфейсов взаимодействия Java-приложений с базами данных.

Поиск рациональных решений, в том числе оптимальная схема построения НПЗ, оптимальный про-

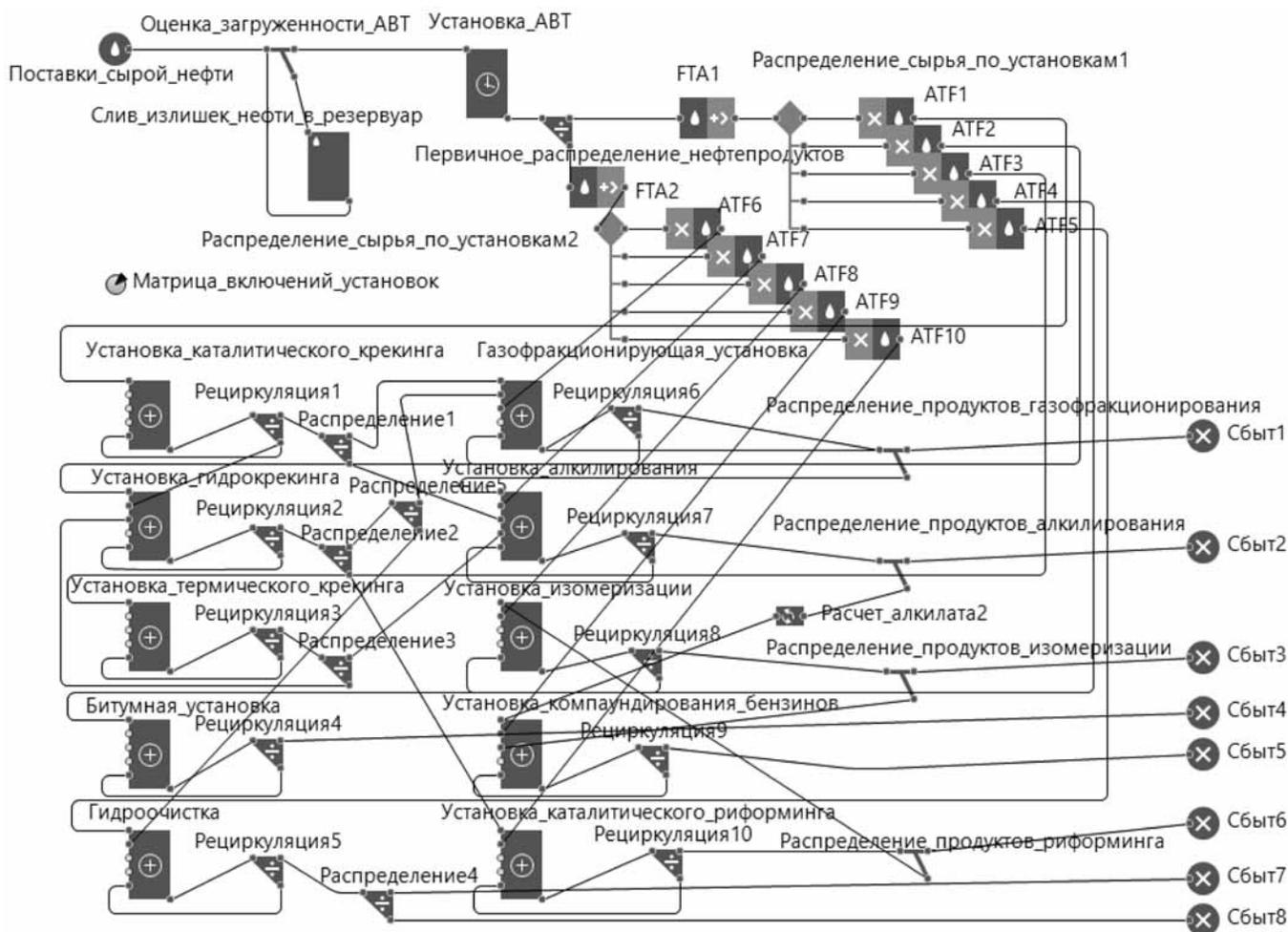


Рис. 3. Фрагмент имитационной модели НПЗ с реализацией в AnyLogic

изводственный план и др. осуществляется посредством генетического оптимизационного алгоритма, встроенного в AnyLogic и бесшовно агрегированного с имитационной моделью НПЗ по целевому функционалу (прибыли).

Выполнение численных, в том числе оптимизационных экспериментов, осуществляется с использованием специальной панели управления, реализованной в виде веб-интерфейса и взаимодействующей с имитационной моделью (AnyLogic) посредством спроектированных веб-сервисов (рис. 4, см. вторую сторону обложки).

Подобные веб-сервисы реализованы с использованием технологии JAX-WS (Java API for XML Web Services), позволяющей реализовывать синхронные и асинхронные взаимодействия. В их числе — передача данных в формате XML (расширяемый язык разметки) посредством SOAP-сообщений (т. е. структурированных сообщений в форме XML-документов с указанием спецификации доступа к веб-сервисам).

Ниже представлен фрагмент Java-кода, обеспечивающего реализацию веб-сервиса по запуску имитационной модели AnyLogic.

```
//Импорт библиотек с API-интерфейсами для программного запуска имитационной модели AnyLogic
import com.xj.anylogic.engine.Engine;
import com.xj.anylogic.engine.connectivity.Database;
import simulation.CustomExperiment;
import simulation.Main;

public applySimulationResponse ApplySimulationModel(final ApplySimulationModelType applySimulationModel)

    throws SimulationModelAssignUWIsFaultFoundFaultMessage{

    //Организация программного доступа к СУБД Oracle
    Locale.setDefault(Locale.ENGLISH);
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Context ic = new InitialContext();
    con = DriverManager.getConnection(connectionUrl, login, pwd);
    con.setAutoCommit(false);

    //Создание потока с запуском имитационной модели AnyLogic
    Thread myThready = new Thread(new Runnable()
    {
        @Override
        public void run() {
            runSimulation(applySimulationModel);
        }
    });
    myThready.start(); //Запуск параллельного потока в асинхронном режиме
}

public void runSimulation(ApplySimulationModelType applySimulationModel){
    {
    //Обеспечение программного доступа к имитационной модели AnyLogic
    CustomExperiment s = new distribution.CustomExperiment(null);
    Engine d = s.createEngine();
    Main m = new Main(d, null, null);
    m.setParametersToDefaultValues();
    d.start(m);
    //Передача параметров подключения к СУБД Oracle в имитационную модель AnyLogic
    m.database = new Database(m, "database", "oracle.jdbc.driver.OracleDriver", connectionUrl, login, password);

    //Инициализация параметров имитационной модели входными данными веб-сервиса
    m. Поставки_сырой_нефти = applySimulationModel.SimulationRequest. Поставки_сырой_нефти;
    .....
    //Программный запуск имитационной модели с заданными значениями параметров
    d.runFast();
    }
}
```

Основным классом в имитационной модели, реализованной в системе AnyLogic, является класс **Main**, который содержит описание всех используемых переменных, параметров и функций модели. Класс **Main** обеспечивает среду функционирования и взаимодействия всех включаемых в модель агентов, например, агентов-продуктов. Вместе с тем для проведения численных экспериментов в AnyLogic с нестандартным пользовательским сценарием и без использования встроенного графического интерфейса, т. е. для программного управления моделью внутри сервиса (на уровне так называемой "backend-функциональности") используется специальный класс **CustomExperiment**.

3. Результаты численных экспериментов

Для проведения численных экспериментов были использованы тестовые (ретроспективные) данные, полученные от крупного НПЗ на условиях соглашения о неразглашении первичной информации NDA (*non-disclosure agreement*). Вместе с тем используемые исходные данные, в том числе по объемам поставок сырой нефти, объемам переработки на уровне каждой установки и др., полностью описывают характеристики реального производственного процесса. Полученные результаты имитационного моделирования были сопоставлены с фактическими результатами деятельности НПЗ в рамках стандартной процедуры верификации и валидации предложенной модели.

Численные эксперименты проводились для двух сценариев, существенно различающихся по характеристике спроса на продукты преимущественно первичной (первый сценарий) и вторичной (второй сценарий) переработки соответственно. Описание исследуемых сценариев представлено в табл. 2.

На рис. 5 (см. третью сторону обложки) представлена динамика, прогнозирующая суточную прибыль НПЗ. Такая динамика формируется с использованием предложенной имитационной модели для опи-

санных выше сценариев и двух изучаемых режимов функционирования системы: до и после оптимизации параметров производственного процесса НПЗ по характеристике прибыли на временном интервале один календарный месяц.

Как следует из данных рис. 5, основной эффект от оптимизации характеристик производственного процесса достигается при сценарии 2, т. е. в условиях преобладающего спроса на продукты вторичной переработки, характеризуемой высокой добавленной стоимостью (высокооктановые бензины, пропан-бутан, полипропилен, синтетические масла, ароматизаторы и др.).

На рис. 6 (см. третью сторону обложки) показана структура объемов производства и поставок нефтепродуктов, соответствующая сценарию 2 до и после оптимизации основных характеристик производственного процесса. Такая оптимизация выполняется с использованием генетического алгоритма, агрегированного с предложенной имитационной моделью НПЗ.

Структуре производства и поставок нефтепродуктов, представленной на рис. 6, *б*, соответствует режим управления, при котором все перерабатывающие установки задействованы и включены в схему построения НПЗ, т. е. $\gamma_i = \tilde{\gamma}_j = 1$ для всех $i \in I, j \in J$. При этом выходные потоки установок первичной переработки (в частности, АВТ, термического и каталитического крекинга и др.) направляются преимущественно на вход установок вторичной переработки (в частности, установки компаундирования бензинов, газофракционирующей установки и др.), средние значения — $\alpha_i = 0,25, \tilde{\alpha}_j = 0,75, i \in I, j \in J$.

Из данных рис. 6 (см. третью сторону обложки) следует, что после проведения оптимизации структура конечной продукции может сильно измениться. Вместе с тем фактическое изменение характеристик производственного процесса, обеспечивающее соответствующую перестройку структуры конечной продукции, может быть ограничено имеющимися контрактными обязательствами. Поэтому внедрение

Таблица 2

Исследуемые сценарии и основные допущения

Сценарии	Основные допущения
<i>Сценарий 1</i>	Низкий целевой уровень глубины нефтепереработки (65 %). Основной спрос на продукты первичной переработки: низкооктановые бензины, битум, мазут, дизельное, реактивное и остаточное топливо с жесткими ограничениями по датам поставки готовой продукции. Дефицит производственно-промышленного персонала (max число не более 1,5 тыс. чел.). Среднеинтенсивный уровень поставок сырой нефти (3...5 тыс. тонн в сутки). Относительно низкая стоимость конечной продукции
<i>Сценарий 2</i>	Высокий целевой уровень глубины нефтепереработки (80 %). Основной спрос на продукты вторичной переработки: высокооктановые бензины, пропан-бутан, полипропилен, синтетические масла, ароматизаторы и др. Отсутствие дефицита производственно-промышленного персонала (max число 2,5 тыс. чел.). Высокоинтенсивный уровень поставок сырой нефти (6...10 тыс. тонн в сутки). Относительно высокая стоимость конечной продукции

полученных с использованием модели результатов реализуется по окончании текущего производственно-сбытового цикла.

Заключение

Представлена новая имитационная модель нефтеперерабатывающего предприятия со сложной схемой взаимодействия оборудования. Для реализации подобной модели впервые применяются гибридные методы агентного и дискретно-событийного моделирования. Предложен подход, обеспечивающий интеграцию имитационной модели НПЗ, реализованной в AnyLogic, с другими системами. В частности, описана интеграция с информационным хранилищем данных, с генетическим оптимизационным алгоритмом и др. Такой подход позволяет, в частности, осуществлять программный запуск модели посредством разработанных веб-сервисов. Программная реализация имитационной модели НПЗ в системе AnyLogic обеспечивает возможность управления важными характеристиками производственного процесса. Например, реализуются "включение и отключение" различных перерабатывающих установок в схему построения НПЗ, изменение долей продукции, распределяемых между системой сбыта и установками вторичной переработки. Варьируется интенсивность загрузки перерабатывающего оборудования сырьем, структурой производственно-промышленного персонала, распределяемого между установками, и др. Сформулирована и решена важная оптимизационная задача НПЗ, целевым функционалом которой является прибыль, накопленная за определенный период (месяц). Проведены численные эксперименты, подтверждающие высокую эффективность результата оптимизации характеристик производственного процесса в условиях преобладающего спроса на продукты вторичной переработки, характеризующейся высокой добавленной стоимостью.

Разработанная модель может быть использована в качестве системы оптимального производственного (суточного) планирования для определения более рациональной и эффективной схемы построения НПЗ. При этом целевыми установками являются рациональное распределение (диспетчеризация) потоков сырья и продуктов переработки между системой сбыта и производственными установками и др.

Дальнейшие исследования на этом направлении планируются направить на разработку детализированной имитационной модели НПЗ, обеспечивающей возможность управления отдельными параметрами, определяющими технологические режимы работы каждой производственной установки. При этом подобная детализированная система управления НПЗ может быть построена без моделирования сложных физико-химических процессов, при условии, что связь между параметрами сырой нефти, внутренними характеристиками производственных установок и др. является экзогенной, в том числе основанной на ранее проведенных лабораторных исследованиях. Вместе с тем в представленной в данной работе укрупненной имитационной модели НПЗ

влияние технологических параметров и характеристик сырой нефти не учитывается, так как производственная система рассматривается в устоявшемся режиме, в котором зависимости между поставками сырой нефти и объемами производства являются линейными и определяются только схемой построения НПЗ и общей производственно-сбытовой стратегией предприятия.

Список литературы

1. **Nikoo M. B., Mahinpey N.** Simulation of biomass gasification in fluidized bed reactor using ASPEN PLUS // *Biomass and Bioenergy*. — 2008. — Vol. 32, No. 12. — P. 1245–1254.
2. **Purohit A., Suryawanshi T.** Integrated Product Blending Optimization for Oil Refinery Operations // *IFAC Proceedings Volumes*. — 2013. — Vol. 46, Iss. 32. — P. 343–348.
3. **Акопов А. С.** Имитационное моделирование. Учебник и практикум для академического бакалавриата. — М.: Юрайт, 2014. — 389 с.
4. **Акопов А. С.** Компьютерная модель нефтеперерабатывающего предприятия // В кн.: *Динамика неоднородных систем. Труды института системного анализа РАН. Ч. 2. Вып. 10.* М.: КомКнига, 2006. — С. 268–280.
5. **Favennec J. P.** Refinery Operation and management // *Series Petroleum Refining*. — 2001. — Vol. 5: Technip. — 434 p.
6. **Брагинский О. Б.** Мировая нефтехимическая промышленность. — М.: Наука, 2003. — 566 с.
7. **Wu N., Bai L., Zhou M., Chu F., Mammari S.** A Novel Approach to Optimization of Refining Schedules for Crude Oil Operations in Refinery // *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. — 2012. — Vol. 42, No. 6. — P. 1042–1053.
8. **Williams H. P.** Model Building in Mathematical Programming. 5th ed. WILEY. 2013. — 432 p.
9. **Власов С. С., Шумихин А. Г.** Моделирование процесса отбензинивания нефти при прогнозировании показателей качества бензина // *Вестник СГТУ*. — 2012. — № 1 (63). — С. 90–94.
10. **Singh A., Forbes J. F., Vermeer P. J., Woo S. S.** Model-based real-time optimization of automotive gasoline blending operations // *Journal of Process Control*. — 2000. — Vol. 10, No. 1. — P. 43–58.
11. **Маряшина Д. Н., Девятков В. В., Марков С. А.** Оптимальное планирование производства нефтепродуктов нефтеперерабатывающего завода // *Труды 9-й Всероссийской научно-практической конференции по имитационному моделированию и его применению в науке и промышленности*. — 2019. — С. 451–456.
12. **Ibrahim A. Y., Ashour F. H., Gadalla M. A.** Refining plant energy optimization // *Alexandria Engineering Journal*. — 2021. — Vol. 60, No. 5. — P. 4593–4606.
13. **Zarandi M. H. F., Ahmadpour P.** Fuzzy agent-based expert system for steel making process // *Expert Systems with Applications*. — 2009. — Vol. 36, No. 5. — P. 9539–9547.
14. **Julka N., Karimi I., Srinivasan R.** Agent-based supply chain management-2: a refinery application // *Computers & Chemical Engineering*. — 2002. Vol. 26, Iss. 12. — P. 1771–1781.
15. **van Dam K. H., Adhitya A., Srinivasan R., Lukszo Z.** Nuances of benchmarking agent-based and equation-based models of an oil refinery supply chain // *In proceedings of 2008 First International Conference on Infrastructure Systems and Services: Building Networks for a Brighter Future (INFRA)*. — 2008. — P. 1–6.
16. **Акопов А. С., Хачатрян Н. К.** Агентное моделирование: уч.-метод. пособие. ЦЭМИ РАН, 2016. — 76 с.
17. **Акопов А. С., Бекларян А. Л., Сагателян А. К.** и др. Система поддержки принятия решений для рационального озеленения города на примере г. Ереван, Республика Армения // *Программная инженерия*. — 2019. — Т. 10, № 2. — С. 87–96.
18. **Макаров В. Л., Бахтизин А. Р., Бекларян Г. Л., Акопов А. С.** Разработка программной платформы для крупномасштабного агент-ориентированного моделирования сложных социальных систем // *Программная инженерия*. — 2019. — Т. 10, № 4. — С. 167–177.
19. **Holland J. H.** Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975. — 183 p.

Simulation Model of an Oil Refinery with a Complex Scheme of Equipment Interaction

A. S. Akopov, akopovas@umail.ru, CEMI RAS, National Research University Higher School of Economics, Moscow, Russian Federation, **N. K. Khachatryan**, nerses@cemi.rssi.ru, CEMI RAS, Moscow, Russian Federation

Corresponding author:

Akopov Adranik S., Professor, Chief Researcher, CEMI RAS, Professor, National Research University Higher School of Economics, Moscow, 105187, Russian Federation
E-mail: akopovas@umail.ru

Received on May 20, 2021

Accepted on June 21, 2021

The article presents the developed simulation model of an oil refinery. The feature of model is the use of agent-based and discrete-event modelling methods. Another feature is applying heuristic optimisation technique to find the best values of key production and resource characteristics of oil refinery equipment interacting within the framework of a complex production scheme to build a refinery plant that provides a deep refining level. At the same time, the objective function of the system is the profit of refinery plant, which takes into account both the sale revenue of petroleum products and operating expenditure. The proposed model is implemented in the AnyLogic system and allows, in particular, maximising the profit of refinery plant at various restrictions, including the volumes of petroleum products supplies required to perform contractual obligations at specified times.

Keywords: simulation modelling of production processes, oil refining, discrete-event modelling, agent-based modelling, optimal production planning, AnyLogic

For citation:

Akopov A. S., Khachatryan N. K. Simulation Model of an Oil Refinery with a Complex Scheme of Equipment Interaction, *Programmnyaya Inzheneriya*, 2021, vol. 12, no. 6, pp. 319–328.

DOI: 10.17587/prin.12.319-328

References

1. **Nikoo M. B., Mahinpey N.** Simulation of biomass gasification in fluidized bed reactor using ASPEN PLUS, *Biomass and Bioenergy*, 2008, vol. 32, no. 12, pp. 1245–1254.
2. **Purohit A., Suryawanshi T.** Integrated Product Blending Optimization for Oil Refinery Operations, *IFAC Proceedings Volumes*, 2013, vol. 46, iss. 32, pp. 343–348.
3. **Akopov A. S.** *Simulation modeling*. Textbook and workshop, Moscow, YURAJT, 2014, 389 p. (in Russian).
4. **Akopov A. S.** Computer model of an oil refinery, *Dynamics of inhomogeneous Systems. Proceedings of the Institute of System Analysis of the Russian Academy of Sciences*. Ch. 2. Issue 10. Moscow, KomKniga, 2006, pp. 268–280 (in Russian).
5. **Favennec J. P.** Refinery Operation and management, *Series Petroleum Refining*, 2001, vol. 5, Technip., 434 p.
6. **Braginsky O. B.** *Global petrochemical industry*, Moscow, Nauka, 2003, 566 p. (in Russian).
7. **Wu N., Bai L., Zhou M., Chu F., Mammari S.** A Novel Approach to Optimization of Refining Schedules for Crude Oil Operations in Refinery, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2012, vol. 42, no. 6, pp. 1042–1053.
8. **Williams H. P.** *Model Building in Mathematical Programming*, 5th ed., WILEY, 2013, 432 p.
9. **Vlasov S. S., Shumikhin A. G.** Modeling the process of oil topping when predicting the quality indicators of gasoline, *Vestnik SSTU*, 2012, no. 1 (63), pp. 90–94 (in Russian).
10. **Singh A., Forbes J. F., Vermeer P. J., Woo S. S.** Model-based real-time optimization of automotive gasoline blending operations, *Journal of Process Control*, 2000, vol. 10, no. 1, pp. 43–58.
11. **Maryashina D. N., Devyatkov V. V., Markov S. A.** Optimal planning for the production of petroleum products of an oil refinery, *Proceedings of the ninth all-Russian scientific and practical conference on simulation and its application in science and industry*, 2019, pp. 451–456 (in Russian).
12. **Ibrahim A. Y., Ashour F. H., Gadalla M. A.** Refining plant energy optimization, *Alexandria Engineering Journal*, 2021, vol. 60, no. 5, pp. 4593–4606.
13. **Zarandi M. H. F., Ahmadvpour P.** Fuzzy agent-based expert system for steel making process, *Expert Systems with Applications*, 2009, vol. 36, no. 5, pp. 9539–9547.
14. **Julka N., Karimi I., Srinivasan R.** Agent-based supply chain management-2: a refinery application, *Computers & Chemical Engineering*, 2002, vol. 26, iss. 12, pp. 1771–1781.
15. **van Dam K. H., Adhitya A., Srinivasan R., Lukszo Z.** Nuances of benchmarking agent-based and equation-based models of an oil refinery supply chain, *In proceedings of 2008 First International Conference on Infrastructure Systems and Services: Building Networks for a Brighter Future (INFRA)*, 2008, pp. 1–6.
16. **Akopov A. S., Khachatryan N. K.** *Agent-based modelling educational methodical manual*, Moscow, CEMI Russian Academy of Science, 2016, 76 p. (in Russian).
17. **Akopov A. S., Beklaryan A. L., Saghatlyan A. K.** et al. Decision support system for the rational greening of the city on the example of Yerevan, Republic of Armenia, *Programmnyaya inzheneriya*, 2019, vol. 10, no. 2, pp. 87–96 (in Russian).
18. **Makarov V. L., Bakhtizin A. R., Beklaryan G. L., Akopov A. S.** Development of software framework for large-scale agent-based modeling of complex social systems, *Programmnyaya inzheneriya*, 2019, vol. 10, no. 4, pp. 167–177 (in Russian).
19. **Holland J. H.** *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975, 183 p.

Я. А. Туровский^{1, 2}, д-р техн. наук, канд. мед. наук, доц., Yaroslav_turovsk@mail.ru,
С. В. Борзунов², канд. физ.-мат. наук, доц., А. А. Вахтин², канд. физ.-мат. наук, доц.,

¹ Институт проблем управления РАН им. В. А. Трапезникова, Москва,

² ФГБОУ ВО Воронежский государственный университет

Повышение частотного разрешения в задачах исследования ЭЭГ на основе ковариационного анализа*

Рассматривается применение метода ковариационного анализа зашумленных сигналов на примере электроэнцефалограммы для увеличения разрешения по частоте применительно к феноменам электрической активности мозга. В первую очередь, метод ориентирован на сигналы, представленные в цифровой форме. Показана возможность увеличения разрешения по частоте по сравнению с методами преобразования Фурье и вейвлет-преобразования.

Ключевые слова: корреляция, ковариационный анализ, электроэнцефалограмма, преобразование Фурье, вейвлет-преобразование, вейвлет Морле, модельный сигнал, спектр мощности

Введение

Одной из главных задач цифровой обработки сигналов, требующей активного использования современных программно-аппаратных средств и эффективных алгоритмов, является задача надежного распознавания паттернов биомедицинских сигналов, в частности, электроэнцефалограмм (ЭЭГ), которая имеет широкий спектр применения в фундаментальных исследованиях [1–4]. При этом важна как временная, так и частотная локализация исследуемых феноменов. Последняя представляет значительный интерес в задачах определения частотной перестройки сигналов от мозга в условиях периодической внешней стимуляции. Как известно, в силу ограничений, присущих классическим методам анализа сигналов, разрешение по частоте и разрешение по времени не могут быть выбраны сколь угодно большими одновременно [5, 6]. Более того, в подавляющем большинстве экспериментов длительность анализируемого сигнала и, как следствие, число дискретных отсчетов, зафиксированных и сохраненных цифровым прибором, имеют связанные с особенностями эксперимента ограничения. Конечно, повышение частоты дискретизации может в некоторых случаях предоставить возможность для более детального анализа. Однако при этом не всегда присутствует техническая и/или экономическая возможность повышения ключевых характеристик используемых приборов [7].

Анализ сигналов в области биологических и медицинских наук требует как разработки новых, так и совершенствования уже существующих алгоритмов цифровой обработки сигналов, имеющих прикладную медицинскую направленность. Отметим повы-

шенные требования к содержательной, физиологической интерпретации полученных данных, позволяющих, в частности, делать выводы относительно детальных характеристик.

Рассмотрим стоящую перед цифровой обработкой сигнала задачу на примере хорошо известного феномена дрейфа альфа-ритма ЭЭГ (записи электрической активности мозга). С учетом того, что данный процесс может занимать время менее чем 0,5 с (фактически достаточно одной волны альфа-ритма, т. е. период от 1/7 до 1/14 с), при этом допускается сдвиг частот в пределах 0,1 Гц [8, 9], то, очевидно, на примере классического Фурье-преобразования возникает противоречие. С одной стороны, для необходимого разрешения по частоте требуется не менее 10 с процесса регистрации ЭЭГ. С другой стороны, очевидно, что эпоха анализа намного превосходит период изменения значений пиковой частоты сигнала, что делает задачу практически нерешаемой с применением этого метода. Наличие частотно-временной развертки на основе вейвлет-анализа делает попытку снять подобное противоречие. Однако, с одной стороны, масштабирование материнского вейвлета позволяет детализировать информацию, представленную одним и тем же числом реализаций процесса, с другой стороны, это усложняет оценку разночастотных колебаний, так как более низкие частоты рассчитываются на больших отрезках времени, чем то же число осцилляций для высоких частот [10–12]. При этом вейвлет Морле, представляющий собой гармоники, модулированные функцией Гаусса, очевидно, должен иметь худшее разрешение по частоте, чем полученные на этом же временном отрезке результаты преобразования Фурье. Таким образом, актуальным представляется поиск подходов и разработка метода анализа сигналов, позволяющего улучшить разрешение по частоте для дискретных временных рядов биомедицинской природы [13, 14].

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-29-01156_мк.

Настоящая работа посвящена применению метода ковариационного анализа зашумленных сигналов для достижения цели повышения разрешения по частоте исследуемого сигнала.

1. Ковариационный метод на основе сдвигов референтных функций

Очевидно, что преобразование Фурье при условии центрирования сигнала и нормирования его на среднее квадратическое отклонение имеет формулу, аналогичную с корреляционным анализом. Таким образом, ковариационный анализ является аналогом преобразования Фурье, при условии нулевого среднего для функций, между которыми рассчитывается коэффициент ковариации.

Как известно, в основе ковариационного анализа сигналов отражена мера степени сходства и различия различных сигналов или сдвинутых по времени копий одного сигнала. Рассмотрим множество гармонических функций $\{f_k(t)\}$, $k = 1, 2, \dots, m$, с различными частотами — референтных функций, определенных на временном диапазоне $[T_0, T]$ следующим образом: $f_k(t) = \sin[\omega_k(t - T_0)]$, причем в начальный момент времени T_0 выполняются равенства $f_k(T_0) = 0$ (рис. 1).

Обозначим исследуемый сигнал, представленный в виде серии дискретных отсчетов $x_i = x(t_i)$, $i = 1, 2, \dots, N$, через x . Упорядоченное множество нормированных и центрированных отсчетов образует вектор \tilde{x} . Суть предлагаемого в работе метода заключается в вычислении значений ковариации вектора \tilde{x} и векторов значений референтных функций f_{ω_k} :

$$C_{\omega_k} = \text{cov}(\tilde{x}, f_{\omega_k}),$$

где $\text{cov}(x, y) = M[(x - M[x])(y - M[y])]$, $M[x]$ — математическое ожидание. В качестве условия разделения близких частот ω_a и ω_b в спектре сигнала $x(t)$ примем:

$$\exists k \in [1, m] : a < k < b, (C_{\omega_a} > C_{\omega_k}) \& (C_{\omega_k} < C_{\omega_b}).$$

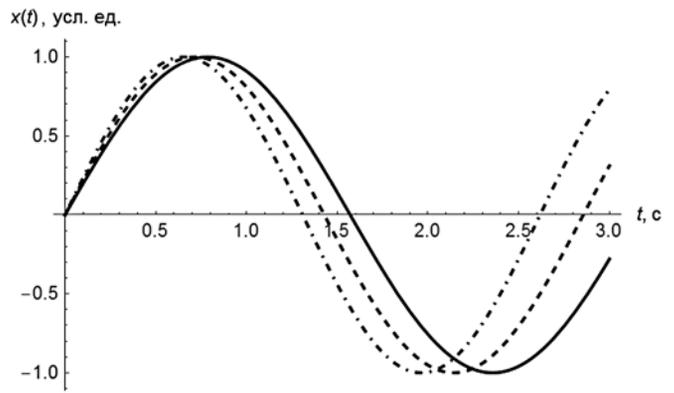


Рис. 1. Элементы множества референтных функций для значений параметров $T = 3$, $\omega = 2, 0, 2, 2, 2, 4$ Гц (сплошная, штриховая и штрих-пунктирная линии соответственно)

За счет уменьшения малого параметра $\delta = \omega_k - \omega_{k-1}$ разрешение по частоте $\Delta f = \delta/(2\pi)$ (Гц) может быть увеличено до требуемых в рассматриваемой задаче значений, при этом разрешение по времени является величиной постоянной.

Рассмотрим полученный результат на примере увеличения разрешения по частоте при конечной длине сигнала, представляющего собой сумму двух близко-расположенных синусоидальных функций, так, что классическое преобразование Фурье не в состоянии их разделить. В качестве модельного примера выберем функцию $d(t) = \sin(2\pi\omega_1 t) + \sin(2\pi\omega_2 t)$ (рис. 2).

Как видно на рис. 3, вид спектра Фурье не позволяет достоверно различить частоты ω_1 и ω_2 за счет близости пиков на значениях 10 и 11 Гц.

Рассмотрим этот же пример для непрерывного вейвлет-преобразования. В качестве базисной вейвлет-функции возьмем действительную часть вейвлета Морле [5, 10] (рис. 4). Выделить исходные частоты на основе вейвлет-преобразования, как видим, в этом случае также невозможно.

В отличие от дискретного преобразования Фурье, Δf зависит от шага смещения по оси времени, что позволяет изменять чувствительность ковариационного метода (рис. 5). Горизонтальные области, простира-

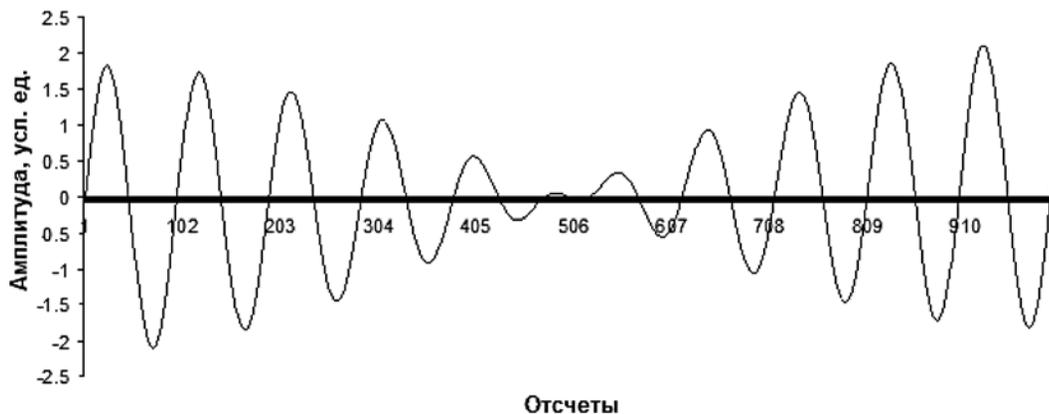


Рис. 2. График модельной функции $d(t) = \sin(2\pi\omega_1 t) + \sin(2\pi\omega_2 t)$ для $\omega_1 = 10$ Гц, $\omega_2 = 11$ Гц. Частота дискретизации равна 1 кГц

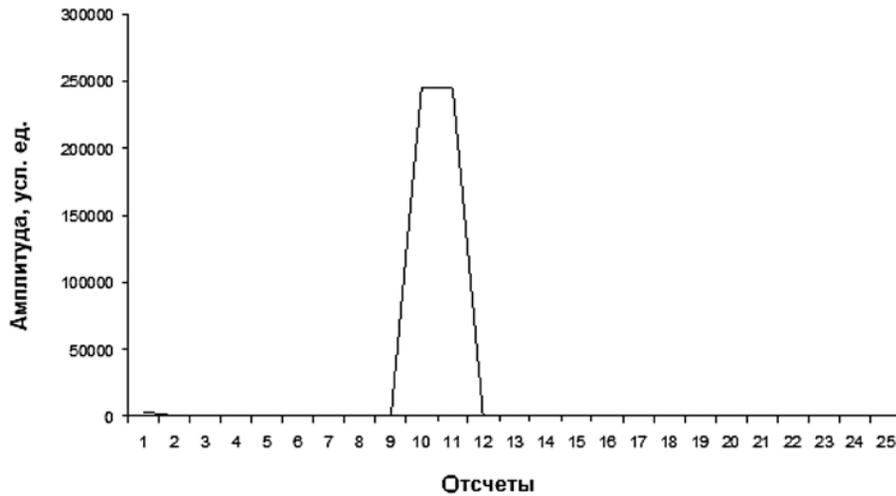


Рис. 3. Дискретный спектр Фурье функции $d(t)$

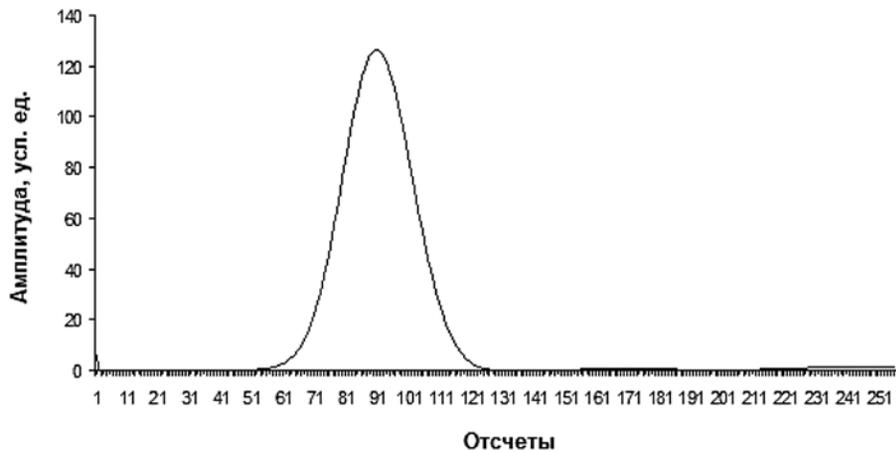


Рис. 4. Спектр дискретного вейвлет-преобразования функции $d(t)$

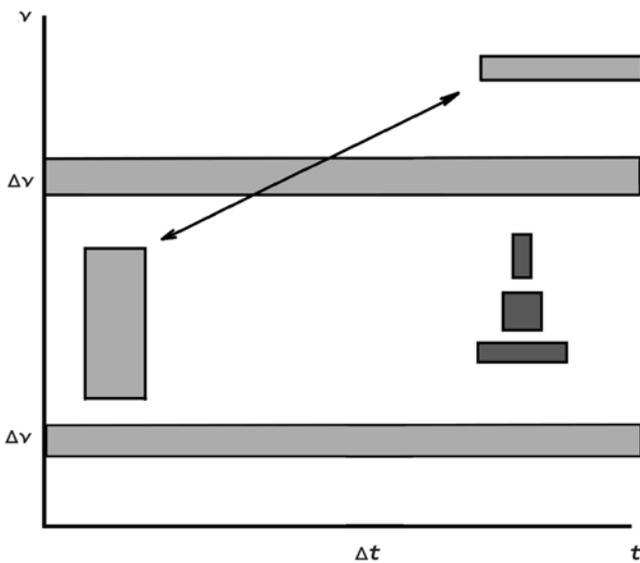


Рис. 5. Пространство "время—частота" и отмеченные на нем области разрешения по времени и по частоте в случае преобразования Фурье и метода ковариационного анализа

ющиеся вдоль всей эпохи анализа $[t_0, T]$, соответствуют преобразованию Фурье. Темные прямоугольники отвечают вейвлет-преобразованию, отмеченные стрелками прямоугольники — ковариационному методу.

На рис. 6 показаны результаты ковариационного анализа модельного сигнала, представленного на рис. 2. Как видно на рис. 6, ковариационный анализ обеспечил разделение двух близкорасположенных гармоник, которое не смогли реализовать другие методы.

2. Влияние оконного преобразования на разрешение сигнала по времени

Введенные в предыдущем разделе референтные функции $f_k(t)$ можно формально рассматривать как применение оконного преобразования вида $\Pi = \theta(t - T) - \theta(t - T_0)$ по отношению к гармоническим функциям, определенным на всей вещественной оси R , где $\theta(t)$ — функция единичной ступени. В ряде случаев медико-биологическому происхождению анализируемого сигнала отвечают гармонические функции, которые подвергнуты иным нелиней-

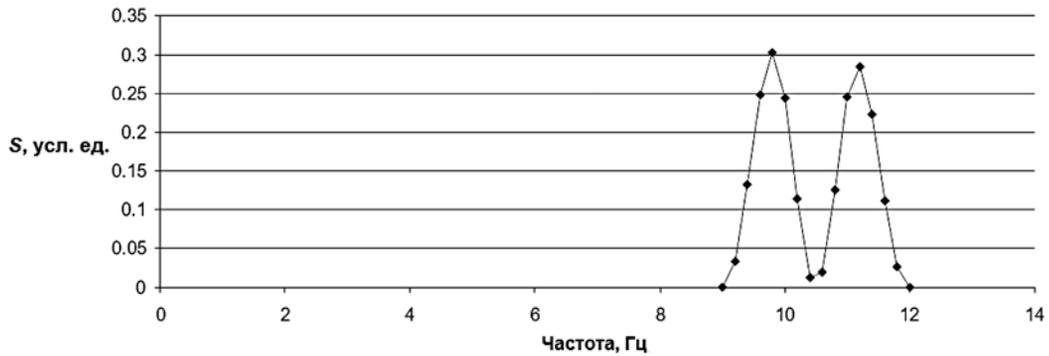


Рис. 6. Спектр мощности S сигнала, полученный с помощью ковариационного метода, демонстрирует достаточное для разделения двух максимумов разрешение по частоте

ным преобразованиям. В частности, действие весовой функции Гаусса $G_{\alpha, \tau}(t) = e^{-\alpha(t-\tau)^2}$ с параметрами $\alpha, \tau \in R$ порождает множество $\{G_{\alpha, \tau}(t)\}$. Использование такого множества приводит к ухудшению разрешения по частоте. Действительно, предположим, что различить сумму двух функций можно при условии отличия их значений в центре симметрии не более, чем в \varkappa раз (рис. 7). На рис. 7 представлено разделение двух сигналов в виде функций Гаусса:

$$e^{-\alpha(t-\tau_1)^2} + e^{-\alpha(t-\tau_2)^2} \leq \varkappa e^{-\alpha(t-(\tau_1+\tau_2)/2)^2}.$$

Подставив в полученную формулу время $t = (\tau_1 + \tau_2)/2$, отвечающее центру симметрии графиков, получим зависимость разрешения по времени через параметры задачи

$$\Delta\tau = |\tau_2 - \tau_1| \geq 2\sqrt{\frac{\ln(2/\varkappa)}{\alpha}}. \quad (1)$$

Из данных на рис. 8 делаем вывод, что, например при $\varkappa = 1$ расстояние между максимумами, которое можно разрешить, зависит от параметра α и равно $\Delta\tau = 1,665$ для $\alpha = 1$, $\Delta\tau = 1,177$ для $\alpha = 2$.

Обратимся к разрешению в частотной области. Поскольку Фурье-образом функции $G_{\alpha, \tau}(t)$ является

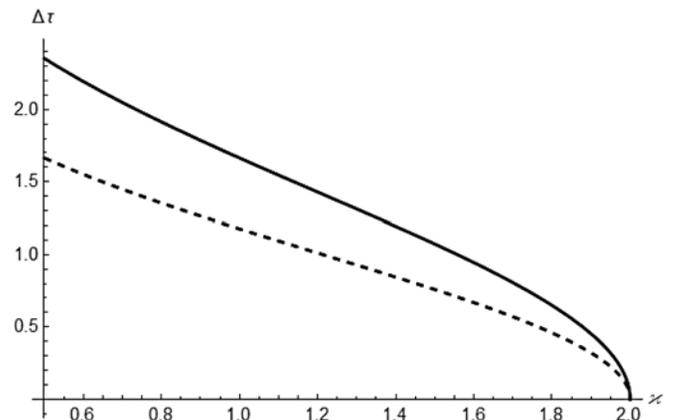


Рис. 8. Зависимость $\Delta\tau$ от параметра \varkappa для $\alpha = 1$ (сплошная кривая), $\alpha = 2$ (штриховая кривая)

$\tilde{G}_{\alpha, \tau}(\omega) = \frac{1}{\sqrt{2\alpha}} e^{-\frac{\omega^2}{4\alpha} + i\omega\tau}$, после вычислений, аналогичных выводу формулы (3), получим:

$$\Delta\omega = |\omega_2 - \omega_1| \geq 4\sqrt{\alpha \ln(2/\varkappa)}. \quad (2)$$

Из формулы (2) видно, что чем больше ширина гауссова окна во временной области, тем меньше ширина его образа в частотном пространстве.

3. Разрешение сигнала по времени для оконной функции в виде гауссовой функции

Вычислим разрешение по частоте для модельной задачи, в которой рассмотрим сумму двух синусоидальных сигналов, умноженную на гауссову функцию (другими словами, проведем частотный анализ сигнала $f(\omega_1, \omega_2, \alpha, \omega)$, "сглаженного" гауссовым кололом, характеризуемым параметром α):

$$f(\omega_1, \omega_2, \alpha, t) = [\sin(\omega_1 t) + \sin(\omega_2 t)]e^{-\alpha t^2}.$$

Для спектра мощности такого сигнала справедлива формула

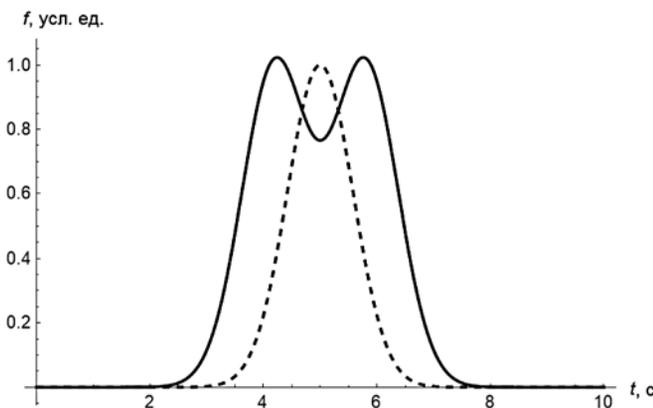


Рис. 7. Пример разделения двух сигналов в виде функций Гаусса

$$S(\omega_1, \omega_2, \alpha, \omega) = \frac{1}{8\alpha} \left[e^{-\frac{(\omega-\omega_1)^2}{4\alpha}} - e^{-\frac{(\omega+\omega_1)^2}{4\alpha}} + e^{-\frac{(\omega-\omega_2)^2}{4\alpha}} - e^{-\frac{(\omega+\omega_2)^2}{4\alpha}} \right]^2.$$

Выясним, при каком условии по спектру мощности имеется возможность определить наличие двух разных частот в заданном сигнале. Для этого вычислим отношение $R(\omega_1, \omega_2, \alpha, \omega)$ амплитуды спектра, взятой в точке на оси симметрии $\omega = (\omega_1 + \omega_2)/2$, к максимальной амплитуде $S(\omega_1)$:

$$R(\omega_1, \omega_2, \alpha, \omega) = \frac{\left[e^{-\frac{(3\omega_1+\omega_2)^2}{16\alpha}} - e^{-\frac{(\omega_1+3\omega_2)^2}{16\alpha}} - 2e^{-\frac{(\omega_1-\omega_2)^2}{16\alpha}} \right]^2}{\left[1 - e^{-\frac{\omega_1^2}{\alpha}} + e^{-\frac{(\omega_1-\omega_2)^2}{4\alpha}} - e^{-\frac{(\omega_1+\omega_2)^2}{4\alpha}} \right]^2}.$$

Частоты разделены, если выполняется условие $R(\omega_1, \omega_2, \alpha, \omega) \leq k$. Полученное нелинейное неравенство относительно параметра α следует решать численными методами. Например, при $\varkappa = 1$ и $\omega_1 = 10$ и $\omega_2 = 11$ минимальное допустимое значение параметра α , при котором возможно разделение частот, равно $\alpha_{\min} = 0,1026$ (рис. 9, на котором продемонстрированы два случая — $\alpha = 0,1$ (частоты разделены), $\alpha = 0,2$ (частоты не разделены)).

4. Разрешение по частоте вейвлет-преобразования

Вейвлет Морле $\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi_0\left(\frac{t-b}{a}\right)$ базируется на материнской функции вида

$$\psi_0(\eta) = \frac{1}{\pi^{1/4}} e^{i\omega_0\eta} e^{-\frac{\eta^2}{2}} \quad (3)$$

введены обозначения для параметров масштаба a и смещения по времени b : $\eta = (t-b)/a$. Параметр ω_0 — частота материнского вейвлета, которая определяет число "осцилляций" функции и обычно принимается равной $\omega_0 = 2\pi$. Поскольку преобразование Фурье (3) имеет вид

$$F[\psi_{a,b}](\omega) = \frac{1}{\pi^{1/4}} \sqrt{ae}^{-\frac{(a\omega-\omega_0)^2}{2} + ib\omega}, \quad (4)$$

и спектр равен $S[\psi_{a,b}](\omega) = |F[\psi_{a,b}](\omega)|^2$, то разрешение вейвлета Морле по частоте при преобразовании двух синусоидальных сигналов может быть оценено с использованием формулы (2). При этом следует учитывать, что в применяющихся на практике численных расчетах обычно применяют дискретное преобразование, соответствующее замене $\omega \rightarrow \frac{2\pi k}{N}$ в формуле (4), где N — число отсчетов

в эпохе анализа, k — порядковый номер отсчета в частотной области. Согласно формуле для спектра $S[\psi_{a,b}](\omega)$, в формулу (2) подставим $\alpha = 1/(4a^2)$ и получим условие разрешимости

$$\Delta\omega = \Delta\left(\frac{2\pi k}{N}\right) \geq 4\sqrt{1/(4a^2)\ln(2/\varkappa)},$$

или

$$\Delta k = \left\lceil \frac{N}{\pi a} \sqrt{\ln\left(\frac{2}{\varkappa}\right)} \right\rceil. \quad (5)$$

В формуле (5) через $\lceil x \rceil$ обозначено ближайшее целое число, не меньшее x (функция *ceil*). Таким образом, например, при $a = 10$ и $\varkappa = 1$ разрешение по частоте равно $\Delta k \geq 15,90 = 16$ (отсчетов), при $a = 20$ и $\varkappa = 1$ разрешение равно $\Delta k \geq 7,95 = 8$ (отсчетов).

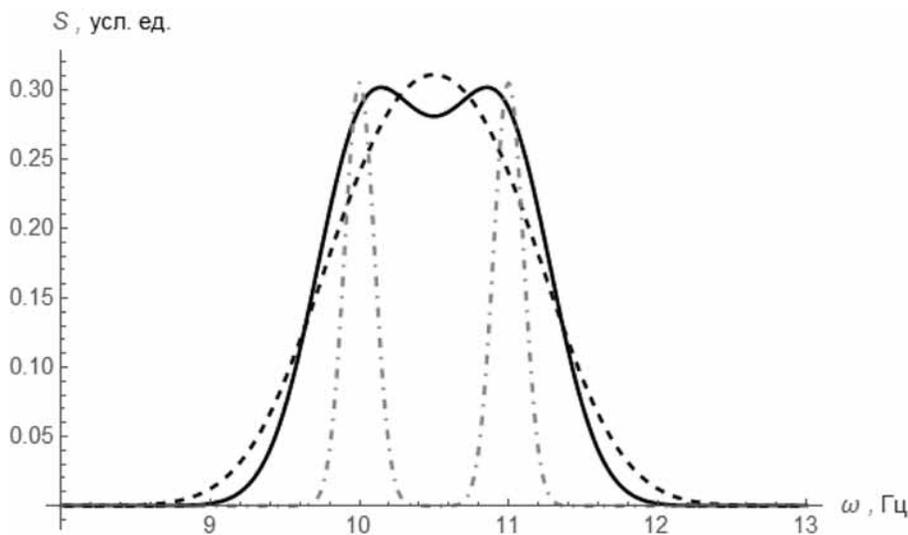


Рис. 9. Зависимость спектра мощности $S(\omega_1, \omega_2, \alpha, \omega)$ от параметра α . Сплошная кривая соответствует значению $\alpha = 0,1$ (частоты разделены), штриховая кривая $\alpha = 0,2$ (частоты не разделены); штрих-пунктирная кривая — нормированный спектр при $\alpha \rightarrow 0$. Значения частот приняты равными $\omega_1 = 10$ Гц и $\omega_2 = 11$ Гц

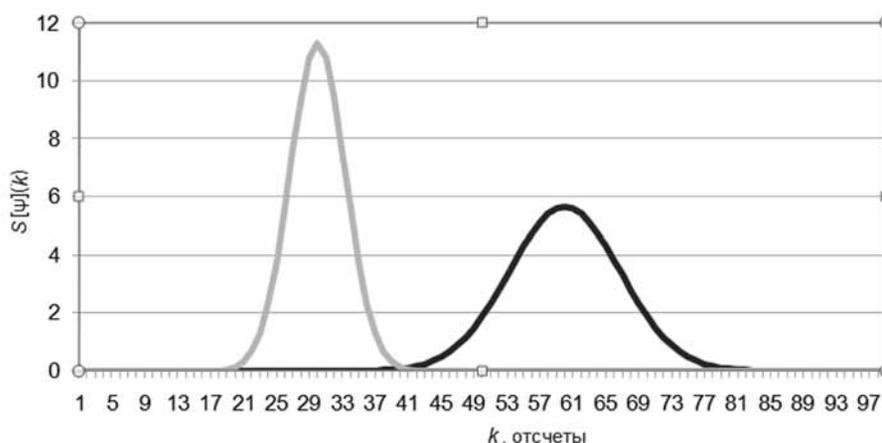


Рис. 10. Зависимость спектров мощности $S[\psi_{10,0}](k)$ (черная кривая) и $S[\psi_{20,0}](k)$ (светлая кривая) от частоты k

На рис. 10 изображены спектры $S[\psi_{10,0}](k)$ и $S[\psi_{20,0}](k)$, демонстрирующие увеличение разрешения при переходе к более детальному масштабу.

В итоге, представив аналитические зависимости для разрешения в частотной области в терминах длительности эпохи анализа T и числа отсчетов N , получаем на рис. 11 разрешение по частоте для классического преобразования Фурье ($\Delta f_{\text{Фурье}} = 1/T$), вейвлет-

анализа $\left(\Delta f_{\text{Морле}} = \left[\frac{N}{\pi a} \sqrt{\ln\left(\frac{2}{\varepsilon}\right)} \right]$ и метода, основан-

ного на ковариационном анализе ($\Delta f_{\text{ковар}} = 1/(NT)$). Заметим, при выбранных значениях $a = 10$ и $a = 20$ область под гауссовой огибающей вейвлета Морле "размещается" на диапазоне эпохи анализа $[0, T]$ с относительной погрешностью не более 1%. Численные значения на рис. 11 для метода, основанного на ковариационном анализе, и для преобразования Фурье, приведены для длительности эпохи анализа $T = 1$ с.

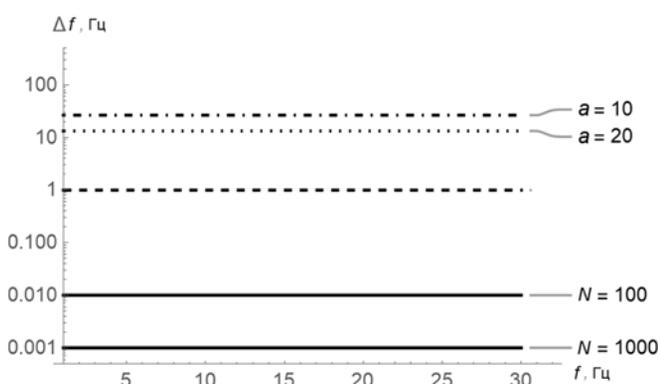


Рис. 11. Разрешение по частоте, достигаемое различными методами: сплошные линии — ковариационный метод ($N = 10^2$ и 10^3); штриховая линия — метод, использующий классическое преобразование Фурье; штрих-пунктирная и пунктирная кривые отражают результаты вейвлет-анализа на основе вейвлета Морле (со значениями параметра $a = 10$ и 20 соответственно). Частота дискретизации равна 1 кГц, длительность эпохи анализа $T = 1$ с (для метода, основанного на ковариационном анализе, и для преобразования Фурье)

Заключение

Представленный в работе метод анализа цифровых сигналов на основе ковариационного анализа демонстрирует возможность увеличения разрешения по частоте по сравнению с классическим преобразованием Фурье и вейвлет-преобразованием с помощью вейвлета Морле на примере модельного сигнала. Следовательно, указанный метод предоставляет потенциальные возможности для решения широкого класса задач в области исследования биомедицинских сигналов, в частности, задач распознавания "паттернов" ЭЭГ; для картирования поверхности головного мозга по данным корреляций активности отдельных участков и многих других.

Список литературы

1. Грубов В. В., Короновский А. А., Ситникова Е. Ю., Храмов А. Е. Частотно-временной анализ характерных паттернов активности нейронных ансамблей головного мозга при помощи непрерывного вейвлетного преобразования // Известия РАН, серия физическая. — 2014. — Т. 78, № 12. — С. 1525–1529.
2. Jones D. L., Touvannas J. S., Lander P., Albert D. E. Advanced time-frequency methods for signal-averaged ECG analysis // J. Electrocardiol. — 1992. — Vol. 25 Suppl. — P. 188–194.
3. Blankertz B., Curio G., Muller K. R. Classifying Single Trial EEG: Towards Brain Computer Interfacing // Advances in Neural Inf. Proc. Systems (NIPS 01). — 2002. — Vol. 14. — P. 157–164.
4. Bai O., Lin P., Vorbach S. et al. Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG // Clinical neurophysiology. — 2007. — Vol. 118, No. 12. — P. 2637–2655.
5. Короновский А. А., Храмов А. Е. Непрерывный вейвлетный анализ и его приложения. — М.: ФИЗМАТЛИТ, 2003. — 176 с.
6. Марпл С. Л. (мл.) Цифровой спектральный анализ и его приложения. — М.: Мир, 1990. — 584 с.
7. Божокин С. В. Вейвлет-анализ динамики усвоения и забывания ритмов фотостимуляции для нестационарной электроэнцефалограммы // Журнал технической физики. — 2010. — Т. 80, № 9. — С. 16–24.
8. Гнездицкий В. В. Вызванные потенциалы мозга в клинической практике. — М.: МЕДпресс-ин, 2003. — 264 с.
9. Yan J., Chen S., Deng S. An EEG-based emotion recognition model with rhythm and time characteristics // Brain Informatics. — 2019. — Vol. 6 (1). — Article no. 7.

10. Астафьева Н. М. Вейвлет-анализ: основы теории и примеры применения // Успехи физических наук. — 1996. — Т. 166, № 11. — С. 1145—1170.

11. Туровский Я. А., Кургалин С. Д., Семенов А. Г. Анализ энцефалограмм на основе исследования цепочек локальных максимумов скейлограмм // Цифровая обработка сигналов. — 2013. — № 2. — С. 20—23.

12. Туровский Я. А., Кургалин С. Д., Семенов А. Г. Динамика цепочек локальных максимумов спектров электроэн-

цефалограмм человека // Биофизика. — 2014. — Т. 59, № 1. — С. 185—190.

13. Пономарев А. В. Основы теории двумерной цифровой обработки сигналов в базисах Фурье с варьируемыми параметрами // Цифровая обработка сигналов. — 2019. — № 2. — С. 12—20.

14. Павлов А. Н., Анищенко В. С. Мультифрактальный анализ сложных сигналов // Успехи физических наук. — 2007. — Т. 177. — С. 859—876.

Increasing Frequency Resolution in EEG Studies based on Covariation Analysis

Ya. A. Turovsky, Yaroslav_turovsk@mail.ru, V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, 117997, Russian Federation; Voronezh State University, Voronezh, 394018, Russian Federation,

S. V. Borzunov, sborzunov@gmail.com, Associate Professor,

A. A. Vahtin, alvahtin@gmail.com, Associate Professor, Voronezh State University, Voronezh, 394018, Russian Federation

Corresponding author:

Turovsky Yaroslav A., Principal Researcher, V. A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences Moscow, 117997, Russian Federation, Head of Lab., Voronezh State University, Voronezh, 394018, Russian Federation

E-mail: Yaroslav_turovsk@mail.ru

Received on May 10, 2021

Accepted on June 07, 2021

The paper discusses the application of the method of covariance analysis of noisy signals using the example of an electroencephalogram to increase the frequency resolution in relation to the phenomena of electrical activity of the brain. The method is focused primarily on signals presented in digital form. The possibility of increasing the frequency resolution in comparison with the method of Fourier transform and wavelet transform is shown.

Keywords: correlation, covariation analysis, electroencephalogram, Fourier transform, wavelet transform, Morlet wavelet, model signal, power spectrum

Acknowledgements: *The reported study was funded by RFBR according to the research project no. 19-29-01156_мк.*

For citation:

Turovsky Ya. A., Borzunov S. V., Vahtin A. A. Increasing Frequency Resolution in EEG Studies based on Covariation Analysis, *Programmnaya Ingeneria*, 2021, vol. 12, no. 6, pp. 329—335.

DOI: 10.17587/prin.12.329-335

References

1. Grubov V. V., Koronovskii A. A., Hramov A. E., Sitnikova E. Y. Time-frequency analysis of characteristic patterns of the activity of neuron ensembles in the brain by means of continuous wavelet transform, *Bulletin of the Russian Academy of Sciences: Physics*, 2014, vol. 78, no. 12, pp. 1242—1245.

2. Jones D. L., Touvanas J. S., Lander P., Albert D. E. Advanced time-frequency methods for signal-averaged ECG analysis, *J. Electrocardiol.*, 1992, vol. 25 Suppl., pp. 188—194.

3. Blankertz B., Curio G., Muller K. R. Classifying single trial EEG: Towards brain computer interfacing, *Advances in Neural Inf. Proc. Systems* (NIPS 01), 2002, vol. 14, pp. 157—164.

4. Bai O., Lin P., Vorbach S. et al. Exploration of computational methods for classification of movement intention during human voluntary movement from single trial, *Clinical Neurophysiology*, 2007, vol. 118, no. 12, pp. 2637—2655.

5. Koronovskii A. A., Khramov A. E. *Continuous wavelet analysis and its applications*, Moscow, FIZMATLIT, 2003, 176 p. (in Russian).

6. Marple S. L. *Digital Spectral Analysis. Second ed.* Dover Publications, Dover Publications, 2019, 432 p.

7. Bozhokin S. V. Wavelet analysis of the dynamics of assimilation and forgetting photostimulation rhythms for non-stationary electroencephalogram, *Journal of Technical Physics*. 2010, vol. 80, no. 9, pp. 16—24 (in Russian).

8. Gnezditskiy V. V. *Evoked Brain Potentials in Clinical Practice*, Moscow, MEDpress-in, 2003, 264 p. (in Russian).

9. Yan J., Chen S., Deng S. An EEG-based emotion recognition model with rhythm and time characteristics, *Brain Informatics*, 2019, vol. 6 (1), Article no. 7.

10. Astafieva N. M. Wavelet analysis: basic theory and some applications, *Physics Uspekhi*, 1996, vol. 39, no. 11, pp. 1085—1108.

11. Turovsky Ya. A., Kurgalin S. D., Semyonov A. G. Analysis of encephalograms based on the study of chains of local maxima of scalograms, *Tsifrovaya Obrabotka Signalov*, 2013, no. 2, pp. 20—23 (in Russian).

12. Turovsky Ya. A., Kurgalin S. D., Semenov A. G. Dynamics of chains of local maxima of spectra of human electroencephalograms, *Biophysica*, 2014, vol. 59, no. 1, pp. 185—190 (in Russian).

13. Ponomarev A. V. Fundamentals of the theory of two-dimensional digital signal processing in Fourier bases with variable parameters, *Tsifrovaya Obrabotka Signalov*, 2019, no. 2, pp. 12—20 (in Russian).

14. Pavlov A. N., Anishchenko V. S. Multifractal analysis of complex signals, *Physics-Uspekhi*, 2007, vol. 50(8), pp. 819—834.




N* Новосибирский государственный университет
 *НАСТОЯЩАЯ НАУКА

Российская Инженерная Академия
 Институт Математики им. С. Л. Соболева СО РАН
 Российская Ассоциация Искусственного Интеллекта
 Новосибирский национальный исследовательский государственный университет
 Institute of Electrical and Electronics Engineers (IEEE)


 Знания
 Онтологий
 Теории

VIII Международная конференция ЗНАНИЯ – ОНТОЛОГИИ – ТЕОРИИ

8 – 12 ноября 2021 г., Новосибирск

Целью конференции является ознакомление с новейшими научными достижениями, обмен знаниями и передовым опытом в области математических методов представления и анализа данных, извлечения знаний и построения теорий предметных областей, анализа формальных понятий и извлечения информации из текстов естественного языка. Сборник трудов конференции будет проиндексирован в РИНЦ, избранные статьи будут проиндексированы в Scopus.

Тематика конференции отражает основные стадии процесса познания:

- Обнаружение закономерностей и извлечение знаний, скрытых в структурированных и неструктурированных данных. Машинное обучение. Распознавание образов, анализ данных. Прогнозирование. Индуктивный вывод.
- Систематизация знаний. Инженерия знаний. Управление знаниями. Извлечение знаний из текстов на естественном языке. Разработка онтологий предметных областей, технологии создания и применения онтологий.
- Построение теорий предметных областей. Разработка семантических и онтологических моделей предметных областей. Анализ формальных понятий. Логическая семантика естественного языка. Нечёткие логики.

Работа конференции планируется в виде пленарных, секционных и стендовых докладов и круглых столов по тематике конференции. Рабочие языки конференции – русский и английский.

Подробности: <http://www.math.nsc.ru/conference/zont/21/>

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская тишина, 23, с. 2, офис 45
 Технический редактор *Е. М. Патрушева*. Корректор *Е. В. Комиссарова*

Сдано в набор 09.07.2021 г. Подписано в печать 23.08.2021 г. Формат 60×88 1/8. Заказ Р1621
 Цена свободная.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".
 119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Рисунки к статье А. С. Акопова, Н. К. Хачатряна

«ИМИТАЦИОННАЯ МОДЕЛЬ НЕФТЕПЕРЕРАБАТЫВАЮЩЕГО ПРЕДПРИЯТИЯ
СО СЛОЖНОЙ СХЕМОЙ ВЗАИМОДЕЙСТВИЯ ОБОРУДОВАНИЯ»

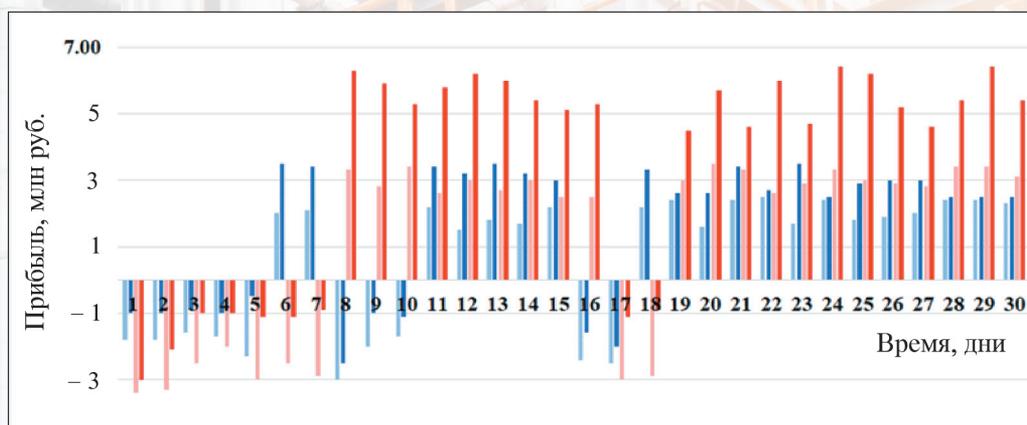


Рис. 5. Прогнозная динамика прибыли НПЗ при различных сценарных условиях:

■ – сценарий 1: до оптимизации; ■ – сценарий 1: после оптимизации;
■ – сценарий 2: до оптимизации; ■ – сценарий 2: после оптимизации;

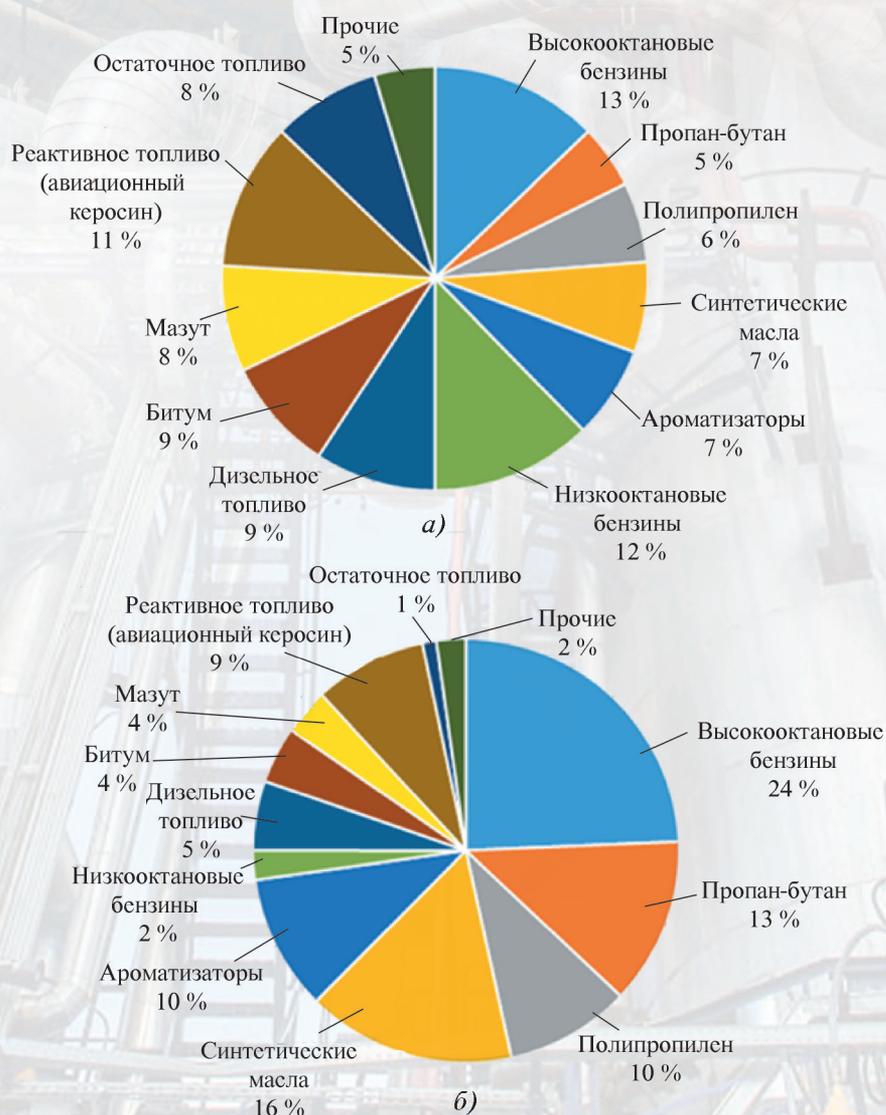
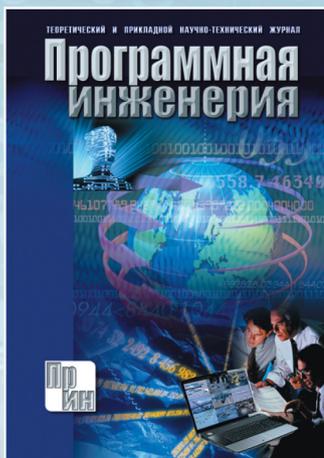


Рис. 6. Структура производства и поставок нефтепродуктов:
а – до оптимизации;
б – после оптимизации

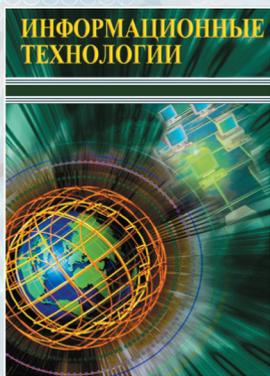
Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



Теоретический и прикладной научно-технический журнал **ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765



Ежемесячный теоретический
и прикладной научно-
технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 72656

Междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопросы разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

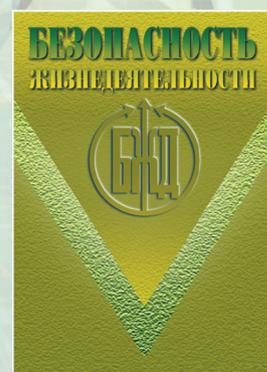
В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:
107076, Москва, Матросская тишина 23 с2, оф. 45. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (916) 392-21-67. E-mail: antonov@novtex.ru