

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24
2018
№ 5

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Барский А. Б., Желенков Б. В., Шаменков Н. А. Высоконадежные управляющие компьютерные сети с нейросетевой адаптивной маршрутизацией на базе перспективных моделей серии «Эльбрус» 291
- Санталов Г. Д., Артемьев Б. В. Беспроводные сенсорные сети с централизованной обработкой данных 299

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Тарасов В. Н., Бахарева Н. Ф., Липилина Л. В. Анализ немарковских сетей массового обслуживания на основе уравнений баланса потоков 306
- Стенников В. А., Барахтенко Е. А., Соколов Д. В. Разработка принципов построения интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики 313

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Акопов А. С., Бекларян А. Л., Хачатрян Н. К., Фомин А. В. Разработка адаптивного генетического оптимизационного алгоритма с использованием методов агентного моделирования 321

БАЗЫ ДАННЫХ

- Зыкин С. В., Зыкин В. С. Коммутативные преобразования в базе данных при редактировании многотабличных запросов 330

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

- Щеглов К. А., Щеглов А. Ю. Моделирование угроз целевых атак 339

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Дам В. Н. OFDM-модуляция в задаче автоматического распознавания вида цифровой модуляции 345

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Коляда А. А., Кучинский П. В., Червяков Н. И. Редукционный метод позиционно-модулярного преобразования больших чисел для нейронных сетей на конечных кольцах 351

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.

ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.

РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.

СОКОЛОВ И. А.,
акад. РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.

ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.

ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.

АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.

ВАСЕНИН В. А., д. ф.-м. н., проф.

ВИШНЕКОВ А. В., д. т. н., проф.

ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В. Г., д. т. н., проф.

ЗАБОРОВСКИЙ В. С., д. т. н., проф.

ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.

ЗАРУБИН В. С., д. т. н., проф.

КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.

КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. В., д. т. н., проф.

ЛЬВОВИЧ Я. Е., д. т. н., проф.

МАРТЫНОВ В. В., д. т. н., проф.

МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.

ПОЛЕЩУК О. М., д. т. н., проф.

САКСОНОВ Е. А., д. т. н., проф.

СОКОЛОВ Б. В., д. т. н., проф.

ТИМОНИНА Е. Е., д. т. н., проф.

УСКОВ В. Л., к. т. н. (США)

ФОМИЧЕВ В. А., д. т. н., проф.

ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

ГРИГОРИН-РЯБОВА Е. В.

ЛЫСЕНКО А. В.

ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

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.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 24
2018
No. 5

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

COMPUTING SYSTEMS AND NETWORKS

Barsky A. B., Zhelenkov B. V., Shamenkov N. A. Highly Reliable Control Computer Networks with Neural Network Adaptive Routing Based on Prospective Models of the Elbrus Family 291

Santalov G. D., Artemiev B. V. Wireless Sensor Networks with Centralized Data Processing 299

MODELING AND OPTIMIZATION

Tarasov V. N., Bakhareva N. F., Lipilina L. V. Analysis of Non-Markov Queuing Networks on the Basis of the Equations Balance of Flows 306

Stennikov V. A., Barakhtenko E. A., Sokolov D. V. A Methodology for the Creation of an Integrated Graphical Environment for Computer Modeling of Energy Pipeline Systems 313

INTELLIGENT SYSTEMS AND TECHNOLOGIES

Akopov A. S., Beklaryan A. L., Khachatryan N. K., Fomin A. V. Development of an Adaptive Genetic Optimization Algorithm using Agent Modeling Methods 321

DATABASE

Zykin S. V., Zykin V. S. Commutative Conversion in the Database when Editing a Multitable Query 330

INFORMATION SECURITY

Shcheglov K. A., Shcheglov A. Yu. Targeted Attacks Threat Modeling 339

DIGITAL PROCESSING OF SIGNALS AND IMAGES

Dam Van Nhich. The OFDM Modulation in the Problem of Automatic Modulation Recognition 345

NEUROTECHNOLOGIES

Kolyada A. A., Kuchynski P. V., Chervyakov N. I. Reducing Method of Positional-Modular Converting Large Numbers for Neural Networks to the End Rings . . . 351

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

А. Б. Барский, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,
Б. В. Желенков, канд. техн. наук, доц., e-mail: boriszhv@gmail.com,
Н. А. Шаменков, канд. техн. наук, ст. науч. сотр., e-mail: shna810516@mail.ru,
НИИЦ (г. Москва) ЦНИИ Войск ВКО МО РФ

Высоконадежные управляющие компьютерные сети с нейросетевой адаптивной маршрутизацией на базе перспективных моделей серии "Эльбрус"

Рассматриваются возможности адаптивной пошаговой маршрутизации информационных пакетов в виртуальной централизованной компьютерной сети сложной распределенной системы управления динамическими процессами и объектами в реальном времени. Применение логической нейронной сети для такой маршрутизации обеспечивает максимальную пропускную способность, надежность, скрытность и живучесть сети. Предлагаются методы организации многоканального параллельного сбора данных головным узлом централизованной компьютерной сети при использовании сети Интернет. Даются рекомендации по развитию архитектуры семейства "Эльбрус" и средств сопряжения со средой функционирования.

Ключевые слова: централизованная компьютерная сеть, допустимая виртуализация, адаптивная маршрутизация, логическая нейронная сеть, web-объекты параллельной сборки данных, семейство "Эльбрус"

Введение

При допустимо виртуальном, т. е. скрытном, физически безадресном размещении вычислительных средств оперативного управления сложными, территориально распределенными системами, такими как энергетические, транспортные, беспроводные телекоммуникационные, оборонные и др., а также при реализации распределенных вычислений по *Grid*-технологии предъявляются высокие требования к средствам информационного взаимодействия элементов и объектов таких систем. Эти требования следуют из общих требований высокой производительности, надежности, помехоустойчивости вычислений и обмена данными, скрытности и живучести систем в условиях деградации под влиянием внешних враждебных действий. Виртуализация вычислительных средств обрела большое значение последние десятилетия. Пользователь, создавая свой вычислительный комплекс, оперирует условными, "математическими", виртуальными адресами компьютеров. "Физические" адреса возникают динамически в результате статического или динамического распределения имеющихся или специально выделенных вычислительных средств между пользователями. Эффективность такого

применения вычислительных средств получила признание как при организации виртуального вычислительного ресурса в многопроцессорных вычислительных комплексах "Эльбрус" [1], так и при организации "облачных" вычислений.

Однако перечисленные выше сложные системы управления допускают лишь ограниченную виртуальность. Так, средства управления одним объектом могут быть созданы лишь на основе множества достаточно "близких" компьютеров, а то и внутри одного вычислительного комплекса, ввиду необходимости оперативной (чаще всего жесткой) связи с объектом. По сути, при этом решается вопрос о требуемой производительности и резервировании [2].

Сформируем абстрактную модель предмета исследований, в которой информационное взаимодействие [3] объектов внутри системы осуществляется двумя способами:

1) на основе направленных радиоканалов связи между смежными узлами, образуемыми объектами, согласно технологии беспроводной телекоммуникационной системы связи;

2) на базе сети Интернет, обладающей ограниченными с точки зрения систем реального времени возможностями по оперативности и информационной безопасности, но обеспечивающей высокую

живучесть при виртуализации вычислительных средств оперативного управления системой.

Отметим, что прокладка стационарных магистралей представляется не только дорогостоящей, но, главное, противоречащей модернизации и развитию, динамической реконфигурации, мобильности и живучести системы. Отметим однако, что в настоящее время интенсивно выполняется прокладка магистральных кабельных систем на основе оптоволокна.

Тогда для рассматриваемых сложных систем справедлива информационно-технологическая сетевая модель, пример которой представлен на рис. 1.

На рис. 1 показано лишь обязательное наличие связей между узлами и внешних связей через сеть Интернет, хотя, как сказано выше, связи между узлами могут быть также реализованы с помощью сети Интернет.

Данная сеть (рис. 1) обладает ведущим и резервными ведущими узлами. Такую компьютерную сеть назовем *централизованной*.

Узлы в этой схеме отражают выделенные из общего вычислительного ресурса компьютеры или компьютерные комплексы (однородные или неоднородные), виртуально, независимо от средств воплощения, связанные между собой одно- или двунаправленными линиями обмена информационными пакетами.

Обмен может совершаться как между отдельными узлами (связь вида "узел → узел"), так и между отдельными узлами и группами узлов (связи вида "узел → много узлов" и "много узлов → узел").

Каждый виртуальный узел жестко связывается с управляемыми средствами системы — стационарными или подвижными.

Считаем, что система связей "каждый с каждым" технически неосуществима на значительной территории, хотя бы из-за требований большой мощности радиоканалов, скрытности связей,

а также трудностей обеспечения синхронной, автономной работы средств системы. Поэтому для согласованной работы системы невозможно избежать транзитивных передач информации. Использование радиоканалов связи для "близких" узлов становится целесообразным, так как они устойчиво работают на сравнительно небольшом расстоянии.

Тогда особую важность обретает разработка *адаптивных алгоритмов маршрутизации* информационных пакетов. Адаптация заключается в динамическом построении маршрута следования каждого отдельного информационного пакета в обход перегруженных или вышедших из строя узлов, а также при прерывании или зашумлении линий или радиоканалов связи.

Однако, как сказано выше, не следует исключать из рассмотрения для сети в целом глобальную сеть Интернет с ее развитыми возможностями обеспечения конфиденциальности и надежности. Для ведущего узла сопряжение с сетью Интернет неизбежно.

Таким образом, на формальном уровне в системе управления реализуются два вида связей для обмена данными:

1. Связи, обеспечивающие распределенные вычисления задач (функциональных модулей) вычислительными средствами системы. Это — сверхоперативные связи, поддерживающие временной режим функционирования системы. Эти связи между узлами в основном отражены на рис. 1.

2. Связи, обеспечивающие системе выход в сеть Интернет. Такими связями обязательно располагают ведущие узлы. Добавим, что организация виртуального управления сложной системой и распределенным решением задач достигается с помощью средств сети Интернет. Значит, все операции управления виртуализацией, а также операции "внешнего общения" также организуются с помощью сети Интернет, как это характерно, например, для центра *Grid*-технологий.

Система связей должна удовлетворять следующим требованиям:

1) обеспечение максимальной пропускной способности как максимального объема циркулирующей информации в единицу времени:

$$W_{\text{обм}} \rightarrow \max;$$

2) обеспечение максимальной вероятности обслуживания запроса на обмен:

$$P_{\text{обм}} \rightarrow \max.$$

Таким образом, сетевые проблемы сложных, территориально распределенных управляющих систем должны решаться на уровне разработки высокопроизводительных вычислительных

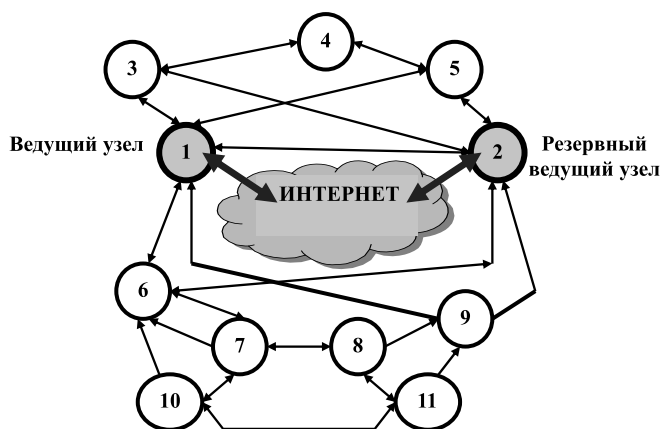


Рис. 1. Примерная схема сетевой централизованной модели системы управления

средств, обеспечивающих их виртуальное размещение и оперативный обмен с помощью беспроводных телекоммуникационных средств, включая сеть Интернет.

1. Нейросетевой адаптивный алгоритм маршрутизации информационных пакетов

Пусть на сети, примерно представленной на рис. 1, задано *отношение смежности* [4]. Информационные пакеты движутся к адресату между смежными компьютерами (узлами). Каждый маршрут по адресу назначения не формируется весь сразу, а реализуется динамически, по шагам, посредством приоритетного обращения к смежным узлам с учетом минимального пути следования к цели, возможного выхода узлов из строя, а также с учетом текущей загрузки этих узлов и текущей оценки качества связи между узлами. Лишь в результате такого комплексного анализа может быть окончательно выбран узел смещения. Таким образом, одновременно реализуемые маршруты в сети оказывают взаимное влияние, и "разводить" их динамически надо так, чтобы избегать пиковых нагрузок узлов и, в конечном итоге, обеспечить максимальную пропускную способность сети, а также максимальную вероятность выполнения передачи.

На каждом i -м узле есть таблица T_i (см. таблицу) предпочтительного смещения при передаче данных на все прочие узлы, кроме смежных.

Здесь R — число узлов сети, в общем случае доступных при передаче пакетов из данного узла через один из смежных, K — число смежных узлов. (В общем случае K — переменная величина.)

При выборе весов ω учитывается территориальное взаимное расположение узлов. Так, очевидно, что приоритетной является та передача, при которой пакет приближается к узлу назначения, хотя в динамике загрузки сети может оказаться, что "круглой" путь ближе "прямого".

В развитие работы [4] введем для всех смежных узлов по отношению к данному узлу *коэффициент расстояния* d_j ($0 \leq d_j \leq 1, j = 1, \dots, K$), значение которого убывает от более "близких" узлов к узлам, более далеким. Ведь очевидно, что

Таблица предпочтительного смещения из данного i -го узла для каждого (не смежного) адреса назначения

Узел (адрес) назначения	Вес смежного узла в направлении передачи			
A_1	ω_{11}	ω_{12}	ω_{1K}
.....
A_R	ω_{R1}	ω_{R2}	ω_{RK}

с ростом расстояния надежность радиосвязи падает. Коэффициент расстояния весьма актуален для движущихся объектов.

Предполагается, что компьютер каждого узла имеет буфер, в котором накапливаются пакеты для дальнейшей отправки. Перегрузка буферов (точнее, достижение критической отметки) должна блокировать прием новых пакетов. Потери информации не предполагаются, пользователь должен быть информирован об этой перегрузке для повторения запроса позже.

Управление передачей пакетов осуществляется с помощью логической нейронной сети, которая первоначально использует для каждого адреса предпочтительные направления передачи пакетов смежным узлам, найденные в таблице. Веса этих смещений используются в качестве весов синаптических связей при обучении сети¹. С помощью обратных связей, осуществляемых смежными узлами, с отрицательными весами передаются коэффициенты k_j загрузки этих узлов, а также (с положительными весами) коэффициенты расстояния d_j и оценки $r_j \leq 1$ ($j = 1, \dots, K$) состояния связи между данным и смежным узлами, которые окончательно влияют на выбор смежного узла для передачи пакета. (Значение k_j следует интерпретировать как достоверность высказывания о том, что буфер узла загружен полностью, значение d_j — как вероятность абсолютной близости j -го узла, а значение r_j — как достоверность высказывания о том, что качество связи между передающим и j -м узлами отличное.) Для каждого конкретного узла составляется своя логическая нейронная сеть, отражающая информацию (в том числе — достижимые адреса), связанную только с этим узлом.

Логическая нейронная сеть, размещенная на одном узле, представлена на рис. 2 (см. вторую сторону обложки). Здесь ω_{ij} — предпочтительные веса смежных узлов при передаче пакета по адресу A_i ; k_j — коэффициент загрузки буфера смежного узла; d_j — коэффициент расстояния; r_j — состояние связи между узлами.

Функция активации нейрона имеет вид $V_j = V_{A_i} \omega_{ij} - k_j + d_j + r_j$, если эта сумма превышает порог h , в противном случае $V_j = 0$.

Порог h выбирается экспериментально так, чтобы предпочтение могло быть выбрано между не полностью загруженными узлами.

Здесь $V_{A_i} = 1$, если A_i ($i = 1, \dots, R$) — адрес назначения информационного пакета. Таким образом, в результате обратной связи максимального возбуждения может достичь совсем не тот смеж-

¹ Логическая нейронная сеть строится обученной, а в процессе эксплуатации легко модифицируется [3].

ный узел, которому первоначально было оказано предпочтение в таблице.

Однако серьезные меры должны быть предприняты во избежание "зацикливания" динамически формируемого маршрута информационного пакета. Для этого пакет сопровождается списком узлов, которые он посетил. Пакет повторно не должен смещаться в узел, в котором он уже был, хотя это не более чем подозрение на возможный цикл. Ведь динамически меняющиеся условия вполне справедливо могли привести к повторному использованию узла. Однако для высокой оперативности передающий узел блокирует (для данного пакета) предложенный ему узел смещения, если адрес такового содержится в списке посещения, и вновь запрашивает у нейронной сети адрес смещения в один из оставшихся смежных узлов.

Необходимо помнить, что для компьютерных вычислений логическая нейронная сеть представляется своей матрицей связей. В рассматриваемом случае однородной логической нейронной сети такая матрица вырождается в таблицу, легко обрабатываемую с помощью *EXCEL*. Так что нейросетевая интерпретация на рис. 2 является лишь теоретической моделью, иллюстрирующей и обосновывающей пошаговый процесс адаптивной динамической маршрутизации.

Более того, в поиске смещения пакета участвует лишь фрагмент нейросети (общей таблицы связей), отображающий движение пакета по данному адресу. Значение возбуждения нейронов, указывающих на смещение, следует искать последовательно, чтобы попутно, за один проход, найти максимально возбужденный нейрон, определяющий решение. Таким образом, алгоритм рассмотренной динамической маршрутизации обладает невысокой полиномиальной сложностью.

Если пакету сместиться некуда, т. е. возбуждение всех нейронов, соответствующих смежным узлам, не превысило порог, пакет "висит" в буфере текущего узла, пока не сдвинется при следующих попытках или пока не иссякнет время его существования (пребывания в буфере). Лишь тогда информационный пакет "теряется".

Отметим, что существующие протоколы динамической маршрутизации могут оценивать только изменение топологии связей и выбирать маршрут на основе пропускной способности. Некоторые протоколы можно настроить на анализ загрузки канала. Но такие протоколы не являются стандартными. Таким образом, для адаптивной пошаговой маршрутизации должен быть разработан новый протокол.

Частными критериями эффективности адаптивной маршрутизации являются: 1) минимум среднего времени выполнения запроса на передачу пакета в сети; 2) минимум времени ожида-

ния отправителем пакета возможности выполнения своих запросов.

Рассмотренный подход легко распространяется на выбор оптимального множества маршрутов совместного следования объектов в транспортной сети любой природы.

Выбор адаптивного алгоритма пошаговой маршрутизации требует глубокого исследования методом имитационного моделирования. В данной статье внимание акцентируется на обосновании требований к вычислительным средствам и к средствам обеспечения надежности.

2. Оптимальная *web*-технология многоканального параллельного приема данных головным узлом

2.1. Проблема параллельного сбора данных на головном узле

Если многоканальные средства ввода-вывода компьютеров интенсивно разрабатываются, то параллельные *web*-технологии практически не изучены.

Распараллеливание обмена вида "много узлов → узел" требует серьезных разработок параллельных каналов приема данных [3]. Ведь если не предпринять никаких мер, то каждый передающий процессор *последовательно* должен связаться с головным процессором другого узла и передать свой пакет данных.

Получила распространение *кластеризация* передающих процессоров узлов. Например, при двухкомпьютерной кластеризации [2, 4] для объединения одиннадцати результатов счета (X_0, \dots, X_{10}) по *SPMD*-технологии на головном 0-м компьютере схема многоярусного "древесного" обмена представлена на рис. 3.

Здесь размерность задачи $n = 11$. Данные распределены для одинаковой обработки по *SPMD*-технологии между $M = 11$ процессорами, в том числе и головным.

Стрелки на рис. 3 соответствуют направлению передачи всех уже имеющихся на процессоре результатов для их накопления, т. е. их объединения с результатами 0-го процессора — адресата. Число ярусов сборки $k = \lceil \log_2 11 \rceil = 4$. Каждый ярус на рис. 3 отображает кластерный список — список процессоров, между которыми формируется обмен на данном уровне, т. е. образуется один кластер.

Пусть $t_{\text{обм}}$ — время единичного обмена в сети, отражающее все необходимые операции согласно протоколу, маршрутизации и пр. Это же время требуется для объединения данных в двухпроцессорном кластере. Тогда полное время T сбор-

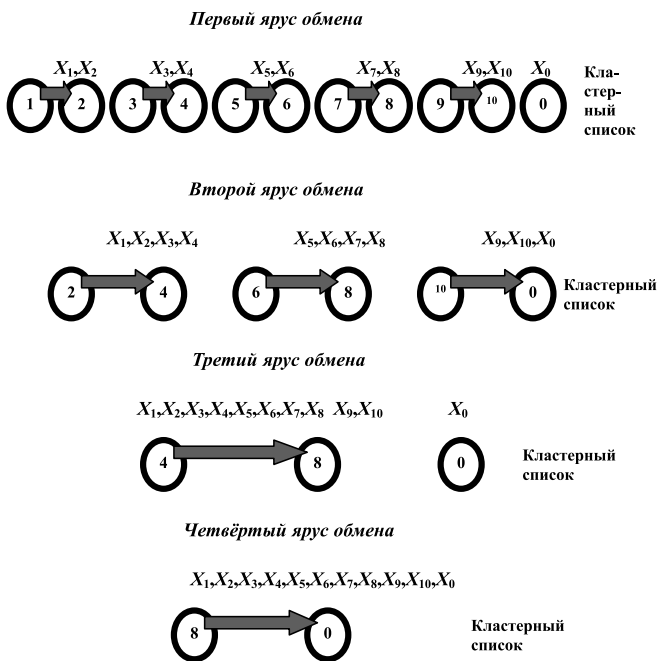


Рис. 3. Сборка результатов счета при двухпроцессорной кластеризации

ки данных на головном процессоре находится по формуле

$$T = \lceil \log_2 11 \rceil t_{обм} = 4t_{обм}.$$

В работе [4] предлагается алгоритм кластеризации процессоров для сборки результатов счета *SPMD*-программы. Этот алгоритм применим в любом случае, когда несколько компьютеров сети приблизительно одновременно закончили расчеты данных, которые необходимо сообщить головному процессору. При этом несомненно, что какие-то накладные расходы требуются для организации этих процессоров в единый пул, совместно осуществляющий кластеризованный обмен. Если такой пул процессоров создать нельзя, процессоры обмениваются по очереди.

В работе [3] рассматривается возможность создания трехпроцессорных элементарных кластеров, если возможно совмещение во времени (распараллеливание) передачи данных лидеру тройки от двух других процессоров (рис. 4).

В этом случае число ярусов сборки резко сокращается. В работе [3] представлен алгоритм сборки результатов с помощью трехпроцессорной кластеризации при выполнении *SPMD*-программы с участием головного процессора.

С ростом числа M процессоров значение $\lceil \log_2 M \rceil$ растет быстрее, чем значение $\lceil \log_3 M \rceil$. Следовательно, при больших значениях M даже последовательный, двойной (непараллельный) обмен внутри кластера может оказаться выгоднее, так как он компенсируется меньшим числом ярусов полной сборки данных.

В работе [3] показано, что время распределенного решения задачи по *SPMD*-технологии существенно зависит лишь от соотношения времени счета одного варианта к времени выполнения сеанса обмена в сети.

Пусть M — число выделенных задаче ресурсов процессоров; n — размерность задачи или число решаемых вариантов; $t_{вар}$ — время счета одного варианта; $t_{сб}$ — время сборки данных в кластере (при двухпроцессорной кластеризации совпадает с $t_{обм}$, при трехпроцессорной кластеризации $t_{сб} \leq 2t_{обм}$). Тогда оптимальные соотношения для выбора M составляют:

$$M \approx 0,7n \frac{t_{вар}}{t_{сб}} \quad \text{— при двухпроцессорной сборке и}$$

$$M \approx 1,1n \frac{t_{вар}}{t_{сб}} \quad \text{— при трехпроцессорной сборке.}$$

Казалось бы, эти соотношения явно не зависят от числа *web*-объектов, используемых для передачи результатов головному процессору (хотя неявно влияют на среднее значение $t_{обм}$). Однако, обобщая стратегию распределенного решения функциональных модулей произвольной управляющей системы, следует с очевидностью признать хотя бы линейную зависимость времени решения задачи от возможностей параллельного многоканального обмена данными, используемыми компьютерами.

Существенным фактором в этом случае является то, что операции обмена образуют *потoki заявок* между компьютерами сети, которые изучаются методами теории массового обслуживания с применением методов имитационного моделирования. Предварительно ясно лишь то, что увеличение числа *web*-объектов в кластере внешнего устройства обмена должно благотворно влиять на сокращение среднего времени обмена в системе управления в реальном времени.

Необходимо признать, что в сетевых технологиях проблема множественного обмена пока решена недостаточно. Проработана быстрая широковещательная и множественная передача по всем или многим адресам. Сборка данных, ког-

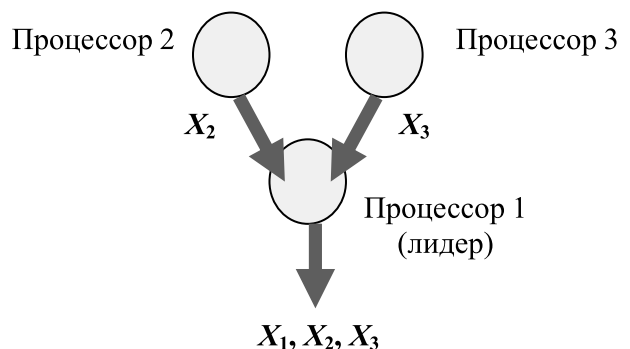


Рис. 4. Параллельный обмен в трехпроцессорном кластере

да многие абоненты передают одному адресату, практически не поддержана ни алгоритмически, ни аппаратно. Но именно ее быстроедействие обусловлено не только широко распространенной сегодня схемой вычислений по *SPMD*-технологии, но и требованиями сложных систем управления в реальном времени.

2.2. Структура оперативной памяти Хост-процессора для многоканальной параллельной сборки данных

Так как головной процессор узла должен обладать существенными особенностями, дающими ему возможность реализовать управление вычислительным процессом и приемом-передачей данных, назовем процессор такого типа *Хост-процессором*.

Структура памяти Хост-процессора узла показана на рис. 5 (см. третью сторону обложки).

Оперативная память (ОП) содержит следующие блоки:

1. Блок собственной ОП, предназначенной только для вычислений и ограниченно доступной другим блокам.

2. N блоков ОП, соответствующих N каналам обслуживания обмена. Каждый блок содержит один или более образов памяти Хост-процессоров передающих узлов, распределенных по каналам обслуживания. Эти образы отражают лишь память этих процессоров, предназначенную для хранения передаваемых результатов вычислений.

3. В качестве внешних устройств используются процессоры памяти $ПП_i, i = 1, \dots, N$, выполняющие функции каналов обслуживания обмена с Хост-процессорами других узлов. Они являются автономными сетевыми объектами (*web-объектами*), обладающими адресами. На основе Интернет-технологии каждый процессор памяти $ПП_i$ связан с закрепленными за ним Хост-процессорами других узлов.

Хост-процессоры узлов независимо (а следовательно, параллельно) по необходимости дублируют свою информацию в этих процессорах, в том числе результаты вычислений. С помощью процессоров памяти выполняется и широковещательный обмен.

Блоки-образы в ОП, ввиду сравнительно небольшого объема, могут быть выполнены по технологии двухвходовой памяти. Однако даже механическое (релейное) их переключение "центральный процессор — сеть" может быть несоизмеримо более оперативным, чем использование кластеризованного сетевого обмена.

Такая структуризация ОП легко согласуется с традиционной блочно-модульной структурой, включающей интерливинг.

2.3. Конвейеризация многоканального приема данных web-процессором памяти от группы абонентов

Пропускная способность сетей связи и передачи данных сегодня достигает значительного объема. Однако в интересах стандартизации и унификации аппаратно-программных средств обмена между *web-объектами*, включая интерфейс пользователя и взаимодействие задач, используется пакетно-кадровая передача. С учетом применения программных средств реальная пропускная способность средств передачи данных в сети снижается на один-два порядка. При решении информационных задач, таких как разработка социальных сетей, торговых площадок, электронной почты, справочных систем и др., на первый план выдвигаются требования высокого сервиса.

Однако для систем управления этого недостаточно. Поток данных на принимающем Хост-процессоре образует "узкое горло", которое необходимо "расшить", используя принципы распараллеливания и конвейеризации. Аналогичная проблема возникает при "древесной" кластеризации, реализуемой для сборки данных в *Grid*-технологии.

Распараллеливание приема данных рассмотрено выше на основе введения нескольких внешних устройств обмена — процессоров памяти, каждый из которых является *web-объектом*, т. е. обладает адресом и достаточной самостоятельностью для выполнения сетевых операций обмена данными.

Второй важный резерв ускорения обмена заключается в применении конвейерной обработки потока данных, поступающих на *web-объект*, перевода их стандартного представления в сети передачи данных в стандартное представление в памяти процессора.

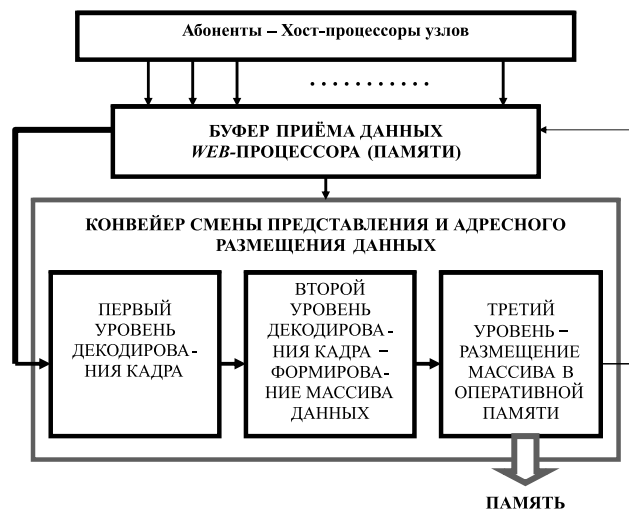


Рис. 6. Конвейеризация приема и размещения данных *web-процессором* памяти

На рис. 6 предполагается, что на первом и втором уровнях конвейера данные переводятся со стандартного *кадрового* представления в линии передачи данных в тип *файл*. На третьем уровне конвейера данные окончательно переводятся в представление задачи, т. е. в тип *массив*, и выполняется адресное размещение массива в ОП.

3. Рекомендации по развитию семейства "Эльбрус"

1. Необходимо продолжить сложившиеся традиции применения средств аппаратного контроля и исправления ошибок, являющихся основным источником информации для построения программных методов построения помехозащищенных управляющих вычислительных процессов.

2. Необходимо развивать парк быстродействующих внешних устройств информационного сопряжения со средствами системы, а также с телекоммуникационными и глобальными компьютерными сетями. Это направление было предусмотрено в предыдущих моделях управляющих МК "Эльбрус".

Еще в 60-х годах прошлого века нашла оправдание идея В. С. Бурцева построения управляющего комплекса М-500 — М-100, в котором процессор М-100 выполнял функции "слабого" резерва и приоритетного обмена по направлениям и объектам. Позднее, в проектах "Эльбрус-1, -2, -3" в обязательную комплектацию входили четыре процессора передачи данных В. И. Перекатова. Указанная тенденция должна быть продолжена, так как требования к производительности управляющего компьютера или компьютерного комплекса, а также требования к оперативности обмена в будущем могут только возрастать.

В этих условиях возлагать задачи обмена на ресурсные процессоры нецелесообразно: ведь в режиме мультипрограммирования два таких процесса высокого приоритета, как счет и обмен, могут мешать друг другу.

Следует помнить о таком новом направлении, как создание быстродействующих *web*-объектов для обмена с сетью Интернет. Это направление приведет к пересмотру и развитию стандартных протоколов обмена.

Таким образом, средства информационного взаимодействия управляющих суперкомпьютеров должны обеспечивать многоканальный быстрый параллельный обмен по технологии беспроводной телекоммуникационной связи, включающей многоканальный доступ в сеть Интернет.

3. Реализованная в МК "Эльбрус" виртуализация ресурса давала превосходный эффект при решении задач распараллеливания по готовности работ в очереди "к процессору". Она же без

дополнительных затрат решала задачу скользящего резервирования. Такой принцип организации вычислений должен быть воспроизведен в перспективных средствах МК "Эльбрус". Однако такой режим виртуализации вычислительного ресурса может потребовать и назначения головного процессора в интересах реализации управляющего вычислительного процесса, так что совмещение централизованного и децентрализованного управления вычислительными ресурсами должно быть доступно.

Отметим, что в интересах контроля и возможности реконфигурации комплекса процессоры МК "Эльбрус-1, -2" были связаны общей шиной и обладали номерами, и пользователь об этом не знал.

4. Применение сетевых технологий вносит особую уязвимость в систему управления перспективными сложными системами вооружения. Представляется, что типизация данных и типовый контроль, воплощенные еще в ранних моделях "Эльбрус", способны эффективно противодействовать современным бурно развивающимся угрозам и атакам противника в попытке взломать и нарушить процесс управления. Здесь необходимо провести самые тщательные исследования на основе предполагаемых применений, понимая, что теговая архитектура, приводящая к существенным излишествам в оборудовании, не является единственным способом воплощения типизации данных.

5. Необходимо развивать принципы аппаратной поддержки языков высокого уровня. По данным испытаний "Эльбрус-1" статистическая оценка показала, что процессор "Эльбрус" в 1,3...1,5 раз быстрее решает задачи, по сравнению с высокопроизводительными ЦВМ, построенными на той же элементной базе, но не обладающими средствами аппаратной поддержки выполнения языковых конструкций. При этом не учитывался стековый механизм выполнения процедур. Идеализация этого механизма в свое время не встретила поддержки разработчика систем управления и является предметом спора до настоящего времени. Известна практика, когда стековый механизм выполнения процедур использовался на этапе программирования управляющих программ, а затем результаты его применения исключались для сведения программы к единственной процедуре.

6. Следует подчеркнуть явную перегруженность операционной системы (ОС) функциями, не применяемыми в ряде систем управления. Повидимому, набор функций ОС должен обосновано выбираться в соответствии с применением ВК.

7. Комплектация моделей семейства "Эльбрус" должна учитывать требования стандартизации и

унификации. Предпочтителен модульный принцип комплектации, допускающий построение ряда моделей, отличающихся по мощности и специализации. Модули исполнительных устройств, сейчас входящие в состав АЛУ, также должны образовывать варьируемые наборы. Их исполнение на микропроцессорной базе дает возможность развития и специализации системы команд, а также реализации процедур ограниченного объема.

8. Необходимо провести самые глубокие теоретические и экспериментальные исследования современных технологий угроз и атак для обоснованной оценки и разработки средств противодействия им в составе ВК "Эльбрус".

Заключение

1. Современные системы управления динамическими процессами и объектами в реальном времени представляют собой территориально распределенные системы с централизованным управлением.

2. Их вычислительные средства образуют централизованную компьютерную сеть, для обеспечения высокой производительности, надежности и живучести использующие принцип допустимой виртуализации. Это определяет специфику средств информационного взаимодействия.

3. Адаптивная маршрутизация информационных пакетов, при которой в пошаговом режиме пакеты движутся в соответствии с динамически

складывающейся загрузкой узлов и состоянием связи между ними, призвана повысить как пропускную способность системы связи, так и вероятность выполнения запроса на обмен. Для обеспечения эффективной работы алгоритма адаптивной пошаговой маршрутизации необходим новый протокол обмена.

4. Компьютеры или компьютерные системы, исполняющие функции управления такой сетью, должны обладать многоканальными средствами беспроводных телекоммуникационных систем, а также средствами многоканального параллельного взаимодействия с сетью Интернет. Для этого предлагается в число внешних устройств управляющего компьютера включить несколько независимых *web*-объектов для приема данных, а также разработать конвейер обработки этого приема.

5. Развитие семейства "Эльбрус" должно учитывать сложившиеся тенденции в поддержке языков высокого уровня, организации вычислений, обеспечении надежности и оперативности средств обмена данными.

Список литературы

1. Ким А. К., Перекатов В. И., Ермаков С. Г. Микропроцессоры и вычислительные комплексы семейства "Эльбрус". СПб.: Питер, 2013. 272 с.
2. Пирожник В. В., Морозов Ю. В., Шаменков Н. А. Обоснование требований к вычислительной технике средств и систем воздушно-космической обороны // Военная мысль. 2015. № 6. С. 51–61.
3. Барский А. Б. Планирование виртуальных вычислений. М.: ИД "ФОРУМ": ИНФРА-М, 2017. 200 с.
4. Барский А. Б. Нейросетевые методы оптимизации решений. СПб.: ИЦ "Интермедия", 2016. 312 с.

A. V. Barsky, Dr. Tech. Sciences, Prof., e-mail: arkbarsk@mail.ru,
B. V. Zhelenkov, Ph. D., Associate Professor, e-mail: boriszhv@gmail.com,
N. A. Shamenkov, Ph. D., Senior Researcher, e-mail: shna810516@mail.ru,
The Russian Transport University (RTU (MIIT))

Highly Reliable Control Computer Networks with Neural Network Adaptive Routing Based on Prospective Models of the Elbrus Family

The possibilities of adaptive step-by-step routing of information packets in a virtual centralized computer network of a complex distributed control system for dynamic processes and objects in real time are considered. The application of a logical neural network for such routing ensures maximum bandwidth, reliability, stealth and survivability of the network. Methods are proposed for organizing multi-channel parallel data collection by the head node of a centralized computer network when using the Internet. Recommendations are given on the development of the architecture of the Elbrus family and interfaces with the environment of functioning.

The main recommendations are as follows:

1. *It is necessary to continue the established traditions of using hardware control and error correction tools to build protected control computing processes.*
2. *It is necessary to develop a fleet of high-speed external interface devices with system resources, as well as with telecommunication and global computer networks. You should use the new direction of creating fast web-objects for exchange with the Internet. This direction will lead to the revision and development of standard exchange protocols.*

3. Virtualization of the resource implemented in ERC "Elbrus" gave an excellent effect when solving the tasks of parallelization on the readiness of work in the queue "to the processor". She also solved the problem of sliding reservation without additional costs. However, the combination of centralized and decentralized management of computing resources should also be available.

4. Application of network technologies introduces a special vulnerability in the management system of advanced complex control systems. It seems that the typing of data and type control, embodied in the early Elbrus models, can effectively counteract today's rapidly developing threats and attacks. Apparently, the tag architecture, which leads to significant excesses in hardware, is not the only way to implement data typing.

5. It is necessary to develop the principles of hardware support for high-level languages. However, reliance on the stack mechanism of performing procedures in its time did not meet the support of the developer of control systems and is the subject of dispute to date.

6. The set of functions of the operating system must be reasonably chosen in accordance with the use of computing facilities.

7. The complete set of models of the Elbrus family should take into account the requirements of standardization and unification. The modular principle of configuration allows the construction of a number of models differing in power and specialization. Modules of actuators must also be composed of variable sets. Their execution on the microprocessor base allows the development and specialization of the command system, as well as the implementation of procedures of limited scope.

Keywords: centralized computer network, permissible virtualization, adaptive routing, logical neural network, web-objects of parallel data collection, Elbrus family

References

1. Kim A. K., Perekatov V. I., Ermakov S. G. *Mikroprocessory i vychislitel'nye komplekсы semeystva "El'brus"* (Microprocessors and computing systems of the Elbrus family), SPb., Piter, 2013, 272 p.

2. Pirozhnik V. V., Morozov Ju. V., Shamenkov N. A. *Obosnovanie trebovanij k vychislitel'noj tehnikе sredstv i sistem vozdušno-kosmicheskoy oborony, Voennaya mysl'* (Justification of

the requirements for the computer technology of means and systems of aerospace defense), 2015, no. 6, pp. 51–61.

3. Barskij A. B. *Planirovanie virtual'nyh vychislenij* (Planning for virtual computing), Moscow, ID "FORUM", INFRA-M, 2017, 200 p.

4. Barskij A. B. *Nejrosetevye metody optimizacii reshenij* (Neural network solutions optimization methods), SPb, IC "Intermediya", 2016, 312 p.

УДК 658.52

DOI: 10.17587/it.24.299-305

Г. Д. Санталов, магистрант, Б. В. Артемьев, д-р техн. наук, проф., boris@artemiev.su, Московский государственный технический университет им. Н. Э. Баумана, Москва, 105005, Россия, e-mail: info@iu.4bmstu.ru

Беспроводные сенсорные сети с централизованной обработкой данных

Представлено перспективное направление использования сенсорных сетей для укрупнения и распространения концепции "Интернет вещей". Основное внимание уделено анализу организации беспроводного массива сенсоров с возможностью централизованной обработки данных. Кратко рассмотрены типы сенсорных сетей, их преимущества и недостатки. Проведен анализ существующих решений и исследование методов обработки данных в различных типах сетей. Наглядно продемонстрированы различные типы сетей. Дана характеристика интерфейсу связи, описание типовых узлов и подборка устройств (в том числе российских). Предложен концепт сенсорной сети, отличающийся отсутствием промежуточного координатора в сети. В результате исследования выявлены характерные аспекты данного концепта, его преимущества и недостатки. Показаны возможности по применению технологии GSM в беспроводных сенсорных сетях в практических условиях, описан механизм взаимодействия устройств и сервера.

Ключевые слова: интернет вещей, сенсорная сеть, беспроводная сенсорная сеть, малая мощность, умные энергосети, мот, координатор, маршрутизатор

Введение

Общество все активнее нуждается в различных автоматизированных системах. Особенно заметна такая тенденция в крупных городах. Именно в мегаполисах развивается технология интернета вещей (Интернет вещей — концепция вычислительной сети физических предметов — "вещей",

оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключаящее из части действий и операций необходимость участия человека), которая все активнее ведет к развитию разнообразных сенсорных сетей.

Интернет вещей [1] — актуальная тема у современных разработчиков ввиду широкого распространения различных электронных устройств в крупных городах. Наличие огромного числа смартфонов, подключенных к сети Интернет, развитие безналичной оплаты и распространение социальных сетей приводят к жизни в едином информационном поле. Это позволяет обмениваться информацией между огромным числом людей. Информация может быть совершенно разной: погода, реклама, новости, личные сообщения и т. д. Такая простота и массовость передачи потоков информации ведет к востребованности разработки систем сбора и обработки данных. Разновидностью таких систем являются сенсорные или датчиковые сети [2—4].

Сенсорные сети можно применять для работы в различных сферах человеческой деятельности: спорт, наука, повседневная жизнь. Беспроводные сенсорные сети, в частности, можно использовать для прогнозирования отказа оборудования в аэрокосмических системах и автоматизации зданий. Вследствие своей способности к самоорганизации, автономности и высокой отказоустойчивости такие сети активно применяют в системах безопасности и военных приложениях. Перспективы использования наблюдаются в сфере контроля здоровья человека, состояния окружающей среды, функционирования производственных и транспортных систем, учета различных ресурсов. Такие широкие возможности активно используют современные фирмы в своих новых разработках. Например, технология iBeacon от Apple должна внести существенный вклад в развитие бизнеса и способность ориентирования в местах с плохим приемом GPS-сигнала [2—4].

Большое потребление энергии является серьезной проблемой для автономной аппаратуры. Сенсорные сети получают наибольшее развитие именно на сегодняшний момент, так как появляются технологии, позволяющие снизить потребляемую устройством мощность. Создаются специальные протоколы связи, ориентированные на пониженное энергопотребление, такие как ZigBee [5—7] и Wibree, а также Bluetooth Low Energy. Это приводит к тому, что снижаются требования к элементам питания, появляется возможность конструирования более компактных, дешевых устройств с большим функционалом.

Современные технологии позволяют создавать сети, состоящие из множества миниатюрных узлов, оснащенных маломощным приемопередатчиком, микропроцессором и сенсором, могут связать воедино глобальные компьютерные сети и физический мир.

При описании различных эффектов были использованы результаты исследования Национального исследовательского университета

"Высшая школа экономики" "Беспроводные сенсорные сети" [4].

Использование сенсорных сетей актуально по ряду причин.

Эффекты:

- снижение уровня отходов от источников питания мотов позволяет снизить негативное влияние на окружающую среду;
- применение самозарядных сенсорных устройств будет способствовать развитию беспроводных устройств интернета вещей, персонализированной медицины (носимые и вживляемые устройства, контролирующие показатели здоровья человека), а также экологически безопасных технологий (*green-technologies*) в энергетике, промышленности (особенно в опасных производствах) и других областях;
- реализация передового производства повысит эффективность, точность и гибкость управления процессами на больших предприятиях и качество выпускаемой продукции, решит вопросы оптимизации ресурсов и определения "узких мест";
- постоянный мониторинг и контроль устройств в энергосистеме снижает количество перебоев в поставках электроэнергии. Повышение рентабельности ее производства приводит к положительным эффектам для промышленности, социальной сферы и окружающей среды (сейчас в России доля энергетики в загрязнении воздуха составляет 26,8 % — максимальное значение в сравнении с другими сферами).

Экономика:

- объем рынка промышленных беспроводных сенсорных сетей в 2017 г. достигнет 3,8 млрд долл.; массовое распространение промышленных беспроводных сенсорных сетей (ИБСС) прогнозируется к 2017—2020 гг.;
- к 2020 г. в мире будет 30,1 млрд беспроводных устройств; потенциальная доля самозарядных устройств достигнет 30...65 %; максимальное проявление тренда прогнозируется в 2020—2030 гг.
- 63 млрд долл. составит к 2020 г. объем мирового рынка технологий "умных" электроэнергетических сетей при среднегодовых темпах роста более 8 %. 2018—2025 гг. — период максимального проявления тренда.

Глобальная проблема нерационального использования электроэнергии особенно актуальна для России. Большие затраты на генерацию электроэнергии увеличивают себестоимость производства продукции, что приходится восполнять конечному потребителю. Для повышения эффективности и надежности энергосистем многие страны переходят к концепции "умных" энергосетей (*smartgrid*).

Такая сеть управляет в режиме реального времени всеми подсоединенными к ней гене-

рирующими источниками, магистральными и распределительными сетями и объектами, потребляющими электроэнергию. Для управления "умной" энергосетью используют беспроводные сенсорные сети, которые контролируют объемы производства и потребления энергии на разных участках. С помощью информационных систем рассчитывается оптимальное распределение энергии в сети, строятся прогнозы на разные сезоны и периоды дня, синхронизируются выработка и транспортировка энергии, контролируется безопасность линий электропередач. Для повышения эффективности энергосети ее некритические элементы на время пониженной активности выключают.

Одним из первых прообразов датчиковой сети можно считать систему СОСУС [8], предназначенную для обнаружения и идентификации подводных лодок. Сенсорные сети стали активно развиваться в середине 1990-х годов, с начала XXI века развитие микроэлектроники позволило производить для таких устройств достаточно дешевую элементную базу.

В 2004 г. в *Scientific American* опубликована обширная статья [9], посвященная интернету вещей, наглядно показывающая возможности концепции в бытовом применении. В статье проиллюстрировано, как бытовые приборы (будильник, кондиционер), домашние системы (система садового полива, охранная система, система освещения), датчики (тепловые, освещенности и движения) и "вещи" (например, лекарственные препараты, снабженные идентификационной меткой) взаимодействуют друг с другом посредством коммуникационных сетей (инфракрасных, беспроводных, силовых и слаботочных сетей) и обеспечивают полностью автоматическое выполнение процессов (включают кофеварку, изменяют освещенность, напоминают о приеме лекарств, поддерживают температуру, обеспечивают полив сада, позволяют сберегать энергию и управлять ее потреблением). Сами по себе представленные варианты домашней автоматизации не были новыми, но упор в публикации на объединении устройств и "вещей" в единую вычислительную сеть, обслуживаемую интернет-протоколами, и рассмотрение интернета вещей как особого явления способствовали обретению концепцией широкой популярности.

Период с 2008 по 2009 г. аналитики корпорации Cisco считают "настоящим рождением интернета вещей", так как, по их оценкам, именно в этом промежутке число устройств, подключенных к глобальной сети, превысило численность населения Земли, тем самым "интернет людей" стал "интернетом вещей".

Первым человеком, который сформулировал работу интернета вещей и, как следствие, беспроводной сенсорной сети был основатель ис-

следовательской группы Auto-ID при Массачусетском технологическом институте (МТИ) Кевин Эштон в 1999 году.

Кевин Эштон родился в Бирмингеме, Великобритания. С 1990 по 1994 г. он читал лекции в университетском колледже Лондона. В 1997 г., работая помощником менеджера в Procter & Gamble (P & G), он заинтересовался использованием RFID для управления цепочкой поставок. Эта работа привела его в МТИ, где он помог создать исследовательский консорциум RFID под названием Auto-ID. Центр был открыт в 1999 г. как спонсируемый промышленностью исследовательский проект в целях создания глобальной открытой стандартной системы для размещения RFID во всем мире. Ученый известен тем, что использует термин "интернет вещей" для описания системы, в которой сеть Интернет подключена к физическому миру через различные датчики [10].

Хотя концепции и идеи сенсорных сетей были сформированы относительно давно, но до сих пор не существует стандартизированной системы для построения датчиковой сети, также как и определенных программно-аппаратных решений. Во многом реализация таких сетей зависит от конкретно решаемой задачи.

Методика построения беспроводных сенсорных сетей с централизованной обработкой данных

Беспроводные сенсорные сети (БСС) в своей основе состоят из компактных вычислительных устройств, называемых мотами. Моты включают в себя датчик или датчики для измерения определенной физической величины (температура, давление, освещенность, шум и др.) и приемопередатчика. Все это позволяет отдельно устройству проводить сбор и обработку необходимых данных, поддерживать связь с внешней средой.

Принятый стандарт IEEE 802.15.4 описывает контроль доступа к беспроводному каналу и физический уровень для низкоскоростных беспроводных личных сетей, т. е. два нижних уровня согласно сетевой модели OSI. "Классическая" архитектура датчиковой сети основана на типовом узле, который включает в себя [11]:

- радиоканал;
- процессорный модуль;
- элемент питания;
- различные датчики.

Типовые узлы представляют собой устройства с определенным набором функций.

Первый вид — устройство с уменьшенным набором функций (англ. *Reduced Function Device*) обладает способностью поддержки топологий "точка—точка", "звезда". Умеет обращаться к координатору сети или маршрутизатору.

Основные характеристики мотов различных производителей

Параметры	Стандарты			
	ML-Node-Z	ML-Node-U	ZigBit	TelosB
	Микроконтроллер			
Процессор	TexasInstruments MSP430		ATmega1281	MSP430
Тактовая частота	От 32,768 кГц до 8 МГц		4 МГц	8 МГц
Оперативная память, Кбайт	10		8	10
Flash-память, Кбайт	48		128	48
	Приемопередатчики			
Тип	IEEE 802.15.4	CypressWireless USBTM LP	IEEE 802.15.4	IEEE 802.15.4
Диапазон частот, МГц	2400...2483,5		2400...2483,5	2400...2483,5
Скорость передачи данных, Кбит/с	250	15,625...250	250	250
Выходная мощность, дБм	-24...0	-35...4	От -28...3	-24...0
Чувствительность, дБм	-95	-93	-101	-90 (min), -94 (typ)
Антенна	Чип		1 или 2 чипа	PCB, SMA
	Внешние интерфейсы			
АЦП	12-разрядный, 7 каналов		10-разрядный, 3 канала	12-разрядный, 7 каналов
Цифровые интерфейсы	I2C/SPI/UART/USB		I2C/SPI/UART/IRQ/JTAG	I2C/SPI/UART/USB
	Прочие параметры			
Напряжение питания, В	0,9...6,5		1,8...3,6	0,9...6,5
Размеры, мм	44 × 33 × 10 мм		19 × 14 × 3	65 × 31 × 6
Температурный диапазон, °С	-40...85	0...70	0...85	-40...123,8

Второй вид — устройство с полным набором функций (англ. *Fully Function Device*) уже может служить в роли координатора сети, поддерживает все типы топологий. Обладает расширенным функционалом сетевого протокола.

Третий вид — сетевой координатор (англ. *Network Coordination Device*) осуществляет глобальную координацию и взаимодействие других устройств сети. В связи с выполняемыми задачами требует наибольшего объема памяти и мощный источник питания.

Возможно использование в типовом узле сенсорной сети в качестве датчика второго передатчика, что позволяет применять датчиковую сеть не только для наблюдения параметров сред и предметов, но и для определения местонахождения и наблюдения передвижений предметов, снабженных специальными радиочастотными метками. Построенная из таких узлов датчиковая сеть образует беспроводную инфраструктуру RTLS.

По причине отсутствия стандартов проектирования сенсорных сетей существует несколько платформ построения мотов (табл. 1).

Датчики в сети взаимодействуют по радиоканалу. Стандарт IEEE 802.15.4 имеет характеристики, представленные в табл. 2.

Вследствие работы мотов в нелицензированном радиоканале могут возникать помехи от по-

сторонних источников радиосигнала. Поэтому необходимо избегать повторной передачи одинаковых данных. А также учитывать, что ввиду внешних воздействий и недостаточной энергоемкости элементы БСС будут выходить из строя навсегда или на какое-то время. Во всех таких случаях схемы обмена данными должны модифицироваться.

Чаще всего мот должен иметь возможность самостоятельно определить свое местоположение, по крайней мере, по отношению к другому моту, которому он будет передавать данные. То есть сначала происходит идентификация всех мотов, а затем уже формируется схема маршрутизации.

Таблица 2

Характеристики радиопередачи данных для IEEE 802.15.4

Полоса частот, МГц	Нужна ли лицензия	Географический регион	Скорость передачи данных, Кбит/с	Число каналов
868,3	Нет	Европа	20	1
902...928	Нет	Америка	40	1...10
2405...2480	Нет	Весь мир	250	11...26

Стандарт ZigBee поддерживает сеть с кластерной архитектурой (рис. 1, см. четвертую сторону обложки). Кластер образуют маршрутизатор и простейшие моты, у которых он запрашивает сенсорные данные. Маршрутизаторы кластеров ретранслируют данные друг другу, и в конечном счете данные передаются координатору. Координатор обычно имеет связь с IP-сетью, куда и направляются данные для окончательной обработки.

Имеется и другой подход к организации сенсорной сети. Он ориентирован на построение одноранговых ячеистых сетей (рис. 2, см. четвертую сторону обложки).

Возможность самоорганизации и самовосстановления сетей ячеистой топологии позволяет в случае выхода части мотов из строя спонтанно формировать новую структуру сети. В любом случае необходим центральный функциональный узел, принимающий и обрабатывающий все данные, или шлюз для передачи данных узлу на обработку.

В такой сети каждый мот способен выполнять функции ретранслятора.

Последний вариант построения БСС можно организовать иным образом.

Концепция работы БСС на GSM

Предлагаемый принцип построения сенсорной сети строится на основе использования GSM-модулей с возможностью выхода в Интернет. Предлагаемая модель представлена на рис. 3 (см. четвертую сторону обложки).

В такой схеме каждому моту необязательно знать о существовании другого. При поломке или систематической потере данных предлагается заменять моты на новые. Причем GSM-модуль можно использовать тот же ввиду его высокой надежности. Проблему разряда батареи отслеживает техник и своевременно устраняет. Установка такой сети предполагается в доступных местах в целях быстрого реагирования и исправления различных ситуаций, приводящих к неработоспособности сети.

Все моты находятся в режиме ожидания. Когда внутренний таймер заканчивает свой отсчет, система переходит в активный режим. В нем она собирает информацию с датчиков и устанавливает связь с сервером. На сервере данные обрабатываются и заносятся в таблицу. После занесения данных и получения подтверждения записи данных на сервере, моты снова переходят в режим ожидания.

При окончании сессии посылается сигнал перехода всех мотов в режим сна, т. е. пониженного энергопотребления. Режим сна можно настроить на определенное время, например, на 2 часа или

1 день. Также существует возможность полного выключения мотов. Данную команду предполагается использовать в случае крайней необходимости, например при полной замене устройства — сенсора, безаварийной замены источника электропитания или при критических поломках.

При разработке концепции работы мота и организации канала связи использовались положения и подходы, изложенные в литературе [5, 6, 12, 13]. Данный вариант построения имеет несколько преимуществ по сравнению с сетями на основе одноранговой ячеистой сети или сети стандарта ZigBee:

- пониженные требования к разработке данной структуры массива сенсоров;
- каждый мот передает данные отдельно и независимо;
- данные передаются сразу на сервер, где обрабатываются необходимым образом.

В ходе работы данной сети могут возникнуть следующие проблемы:

- Зависимость от GSM-сигнала. Сюда же входит зона покрытия оператора, количество средств на сим-карте и стабильность сотового сигнала.
- Повышенное энергопотребление мота. Это связано с тем, что GSM-модуль требует для своего функционирования большое напряжение питания.

Такая сеть не требует выполнения сложного программирования и организации, но обладает широким функционалом.

Прототип мота для данной концепции сети состоит из нескольких основных блоков:

- блок питания устройства;
- датчик температуры;
- датчик влажности;
- микроконтроллер;
- телефон стандарта GSM.

Питание устройства представляет собой три последовательно соединенные батарейки с номинальным напряжением 3 В каждая и током 20 мА. Общее напряжение равно 9 В, а ток 20 мА, $I_{пит} = 20$ мА.

Датчик температуры DS18B20 по данным технической документации потребляет ток 1 мА, $I_{дт} = 1$ мА.

Датчик влажности DHT11 по данным технической документации потребляет ток во время измерения 2,5 мА, $I_{дв} = 2,5$ мА.

Микроконтроллер — ток, потребляемый микроконтроллером Atmega32A согласно технической документации при частоте 1 МГц, напряжении питания 3 В и температуре окружающей среды 25 °С в активном режиме составляет 1,1 мА.

Так как планируется использовать данное устройство при других положительных температурах, а также в холодных условиях (температура до -30 °С), то необходимо ввести параметр, учи-

тывающий изменение тока при иных климатических условиях.

Пусть $\Delta I_{\text{МК}}$ — параметр, учитывающий изменение тока на микроконтроллере. Тогда общий ток на микроконтроллере (в активном режиме)

$$I_{\text{МК}} = I_{\text{ну,МК(a)}} + \Delta I_{\text{МК}} = 1,1 \text{ мА} + 10 \text{ мА} = 11,1 \text{ мА},$$

где $I_{\text{ну МК(a)}}$ — ток, потребляемый микроконтроллером Atmega32A согласно технической документации в нормальных условиях.

Телефон стандарта GSM представлен модулем А6 Thinker, по данным технической документации которого ток в активном режиме находится в пределах 100...900 мА. В режиме экономии — 3 мА, в режиме простоя — 80 мА. При включении модуля ему достаточно напряжения 5 В. В режиме простоя — от 40 до 80 мА (реле открыто). При приеме звонка или отправке СМС — от 80 до 200 мА (реле замкнуто).

Для надежности предположим наибольшее потребление тока в схеме — 900 мА в активном режиме.

$$I_{\text{gsm(a)}} = 900 \text{ мА} — \text{активный режим.}$$

$$I_{\text{gsm(пр)}} = 80 \text{ мА} — \text{режим простоя.}$$

Общий ток в схеме равен сумме токов на основных блоках.

Суммарный ток схемы равен (в активном режиме):

$$I_{\Sigma} = I_{\text{пит}} + I_{\text{дт}} + I_{\text{дв}} + I_{\text{МК}} + I_{\text{gsm(a)}} = \\ = 20 + 1 + 2,5 + 11,1 + 900 = 934,6 \text{ мА}.$$

Максимальная мощность устройства. Максимальная мощность устройства $P_{\text{уст}}$ равна

$$P_{\text{уст}} = I_{\Sigma} \cdot U_{\text{пит}} = 934,6 \cdot 9 = 8,411 \text{ Вт}.$$

Расчет мощности, рассеиваемой в качестве теплоты. Примем, что эффективность передачи антенны и работы устройства $\eta = 80 \%$.

Тогда тепловая мощность, рассеиваемая в пространстве, будет равна:

$$P_{\text{рас}} = P_{\text{уст}} \cdot 0,20 = 8,411 \cdot 0,20 = 1,68 \text{ Вт}.$$

Для конкретной реализации мота выбор представленной элементной базы основывается на возможности работы отдельных блоков устройства при отрицательных температурах до $-25 \text{ }^{\circ}\text{C}$.

Таким образом, проектируемый мот имеет возможность работы в условиях отрицательных температур. При этом стоимость изготовления устройства невелика, сборка проста, используются широко представленные на рынке надежные электронные элементы. Однако остается проблема с оптимизацией энергоэффективности — настройка режимов работы, время автономной работы. Решение этой проблемы во многом зависит от качества настройки программного обеспечения. Правильно настроенные режимы работы

мота и передачи данных позволят продлить срок службы батареи.

На повышенное энергопотребление устройства может также влиять недостаточно стабильный GSM-сигнал. В таком случае возможна замена антенны на более мощную модель или оптимизация ее расположения в пространстве.

Предложенная концепция БСС позволяет создать распределенную недорогую сеть датчиков в условиях отрицательных температур. Такая сеть может быть использована, например, для снятия параметров снега на лыжных трассах, на горнолыжных склонах.

Заключение

Сенсорные сети могут быть использованы во многих прикладных областях. Беспроводные сенсорные сети — это новая перспективная технология, и все связанные с ней проекты в основном находятся в стадии разработки. Уже на этом этапе наблюдается большая польза от применения этой технологии. Ее развитие будет во многом полезно как обычным людям, так и большим компаниям.

Данную технологию можно использовать в различных областях человеческой деятельности. Например, в лыжных видах спорта очень важно определять температуру и влажность окружающей среды. Эти параметры серьезно влияют на выбор различных мазей скольжения, мазей держания и способов обработки лыж. При правильной обработке скользящей поверхности можно добиться существенного улучшения результатов спортсменов, а также повышения комфорта от катания для любителей лыжных гонок. Применение технологий сенсорных сетей в лыжных гонках позволит всем заинтересованным людям получать информацию о состоянии трассы заблаговременно и дистанционно.

С распространением концепции интернета вещей данная технология будет все прочнее и глубже входить в повседневную жизнь каждого жителя мегаполиса.

Отдельные результаты исследований получены в рамках проекта МОН РФ No. 14.579.21.0142 UID RFMEFI57917X0142.

Список литературы

1. **Интернет** вещей. URL: <https://goo.gl/kwyz12> (дата обращения: 22.10.2017).
2. **Vlasov A. I., Yudin A. V., Salmina M. A., Shakhnov V. A., Usov K. A.** Design Methods of Teaching the Development of Internet of Things Components with Considering Predictive Maintenance on the Basis of Mechatronic Devices // International Journal of Applied Engineering Research. 2017. Vol. 12. N. 20. P. 9390—9396.
3. **Буторин П. С., Григорьев П. В.** Система оповещения в автобусном транспорте на основе технологии "Интернета

вещей" // Энергосбережение и эффективность в технических системах. Сборник трудов. Тамбов. 2017. С. 384–385.

4. **Глобальные** технологические тренды // ТРЕНДЛЕТТЕР. Октябрь 2014. № 4. Информационно-коммуникационные технологии. Национальный исследовательский университет "Высшая школа экономики". URL: <https://goo.gl/zjxdKL> (дата обращения: 22.10.2017).

5. **Шахнов В. А., Власов А. И., Резчикова Е. В., Токарев С. В., Смuryгин И. М., Денисенко Н. А., Муравьев К. А.** Способ функционирования беспроводной сенсорной сети. Патент РФ на изобретение. RUS 2556423 05.07.2013.

6. **Краснобрыжий Б. В., Лавров И. В., Муравьев К. А., Чебова А. И.** Анализ беспроводных сенсорных сетей на основе стандарта ZIGBEE // В сб.: Научные технологии и интеллектуальные системы. XV Молодежная научно-техническая конференция, Москва, МГТУ им. Н. Э. Баумана, 2013. С. 306–314.

7. **Зимин Д. В., Муравьев К. А.** Анализ проблем энергоэффективности беспроводных сетей передачи данных на базе стека протоколов ZIGBEE // Труды международного

симпозиума Надежность и качество. Пенза: ПГУ. 2016. № 1. С. 195–197.

8. **СОСУС**. URL: <https://goo.gl/aJBHZV>. (дата обращения: 22.10.2017).

9. **Gershenfeld N., Krikorian R., Cohen D.** The Internet of Things // Scientific American. October 2004. URL: <https://goo.gl/fjPT9z> (дата обращения: 23.10.2017).

10. **Ashton K.** That "Internet of Things" Thing // RFID Journal. URL: <https://goo.gl/Rn8Bzq> (дата обращения: 24.10.2017).

11. **Беспроводной** промышленный мониторинг. Институт точной механики и вычислительной техники им. С. А. Лебедева РАН. URL: <https://goo.gl/jxgGHU> (дата обращения: 24.10.2017).

12. **Власов А. И., Иванов В. В., Косолапов И. А.** Методы упреждающего прогнозирования состояния широкополосной сети связи // Программные продукты и системы. 2011. № 1. С. 3–6.

13. **Власов А. И., Юлдашев М. Н.** Гауссовские процессы в регрессионном анализе состояний беспроводной сенсорной сети // Технологии электромагнитной совместимости. 2017. № 3 (62). С. 35–43.

G. D. Santalov, Magistant, **B. V. Artemiev**, D. Sc., Professor, boris@artemiev.su,
Bauman Moscow State Technical University, Department of Design and Technology of Electronic Devices,
Moscow, 105005 Russia, info@iu4.bmstu.ru

Wireless Sensor Networks with Centralized Data Processing

This article is devoted to the perspective direction of using sensor networks for the enlargement and dissemination of the concept of "Internet of Things". The main attention is paid to the description of the organization of a wireless array of sensors with the possibility of centralized data processing. The types of sensor networks, their advantages and disadvantages are briefly considered. The article analyzes existing solutions and studies data processing methods in various types of networks. Different types of networks are clearly demonstrated. The characteristic of the communication interface is given. Typical units and devices, including Russian ones, are considered. The main parameters of which are given in the table. A concept of a sensor network is proposed, which is distinguished by the absence of an intermediate coordinator in the network. As a result of the research, the characteristic aspects of this concept, its advantages and disadvantages were revealed. As the transmission medium, a radio channel of the IEEE 802.15.4 standard. The possibilities of using GSM in wireless sensor networks in practical conditions are shown, the mechanism of interaction between devices and the server is described.

Keywords: Internet of Things, sensor network, wireless sensor network, low power, smart grid, motes, coordinator, router

References

1. **Internet** veshhej, URL: <https://goo.gl/kwyz12> — (date of access: 22.10.2017).

2. **Vlasov A. I., Yudin A. V., Salmina M. A., Shakhnov V. A., Usov K. A.** Design Methods of Teaching the Development of Internet of Things Components with Considering Predictive Maintenance on the Basis of Mechatronic Devices, *International Journal of Applied Engineering Research*, 2017, vol. 12, no. 20, pp. 9390–9396.

3. **Butorin P. S., Grigor'ev P. V.** Sistema opoveshhenija v avtobusnom transporte na osnove tehnologii "Interneta veshhej", *Jenergoberezhenie i jeffektivnost' v tehnikeskix sistemah*, Sbornik trudov, Tambov, 2017, pp. 384–385.

4. **Global'nye** tehnologicheskie trendy, *TRENDLETTER*, no. 4, *OKTJaBR* 2014, Informacionno-kommunikacionnye tehnologii // Nacional'nyj issledovatel'skij universitet "Vysshaja shkola jekonomiki". Jelektronnyj resurs. URL: <https://goo.gl/zjxdKL> (date of access: 22.10.2017).

5. **Shahnov V. A., Vlasov A. I., Rezchikova E. V., Tokarev S. V., Smurygin I. M., Denisenko N. A., Murav'ev K. A.** Sposob funkcionirovanija besprovodnoj sensornoj seti, Patent RF na izobretenie. RUS 2556423 05.07.2013.

6. **Krasnobryzhij B. V., Lavrov I. V., Murav'ev K. A., Chebova A. I.** Analiz besprovodnyh sensoryh setej na osnove

standarta ZIGBEE, *V sbornike: Naukoemkie tehnologii i intellektual'nye sistemy XV Molodezhnaja nauchno-tehnicheskaja konferencija*. Moscow: MGTU im. N. Je. Baumana, 2013, pp. 306–314.

7. **Zimin D. V., Murav'jov K. A.** Analiz problem jenergojeffektivnosti besprovodnyh setej peredachi dannyh na baze steka protokolov ZIGBEE. *Trudy mezhdunarodnogo simpoziuma Nadezhnost' i kachestvo*, Penza, PGU, 2016, no. 1, pp. 195–197.

8. **SOSUS**. URL: <https://goo.gl/aJBHZV> — (date of access: 22.10.2017).

9. **Gershenfeld N., Krikorian R., Cohen D.** The Internet of Things, *Scientific American*. October 2004. URL: <https://goo.gl/fjPT9z> (date of access: 23.10.2017).

10. **Ashton K.** That "Internet of Things" Thing, *RFID Journal*, URL: <https://goo.gl/Rn8Bzq> (date of access: 24.10.2017).

11. **Besprovodnoj promyshlennyj monitoring**, Institut tochnoj mehaniki i vychislitel'noj tehniki im. S. A. Lebedeva RAN. URL: <https://goo.gl/jxgGHU> (date of access: 24.10.2017).

12. **Vlasov A. I., Ivanov V. V., Kosolapov I. A.** Metody uprezhdajushhego prognozirovanija sostojanija shirokopolosnoj seti svjazi, *Programmnye Produkty i Sistemy*, 2011, no. 1, pp. 3–6.

13. **Vlasov A. I., Juldashv M. N.** Gaussovskie processy v regressionnom analize sostojanij besprovodnoj sensornoj seti, *Tehnologii Jelektromagnitnoj Sovmestimosti*, 2017, no. 3 (62), pp. 35–43.

В. Н. Тарасов, д-р техн. наук, проф., зав. каф., e-mail: veniamin_tarasov@mail.ru,
Н. Ф. Бахарева, д-р техн. наук, проф., зав. каф., e-mail: nadin1956_04@inbox.ru,
Л. В. Липилина, аспирант, e-mail: mila199113@gmail.com,

Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Анализ немарковских сетей массового обслуживания на основе уравнений баланса потоков

Предложены модели математического мультиплексирования и демультимплексирования потоков, а также выражения для определения числовых характеристик распределения интервалов выходного потока из системы массового обслуживания (СМО). Совместно они позволяют записать уравнения равновесия относительно средних значений и дисперсий распределений интервалов времени между соседними требованиями в сетях массового обслуживания при произвольных законах поступления и обслуживания. В качестве узлов сети рассматриваются СМО $M/M/1$ с запаздыванием во времени, $M_2/M_2/1$ и $M_2/M/1$.

При общих допущениях о вероятностных распределениях времени между соседними требованиями в входных потоках и времени обслуживания в узлах предложенный подход позволяет определить средние значения и дисперсии интервалов времени между требованиями всех потоков сети массового обслуживания, а также все основные показатели функционирования таких сетей. Такой подход также можно расширить при необходимости и для моментов 3-го порядка. С точки зрения теории вероятностей так будет точнее, чем расчет на уровне двух моментов распределений.

В условиях неполной информации о законах распределений потоков предложенный подход к анализу производительности немарковских сетей массового обслуживания является на данный момент приемлемым.

Ключевые слова: сети массового обслуживания, системы массового обслуживания $M/M/1$, $M_2/M_2/1$, $M_2/M/1$, агрегирование и разрежение потоков

Введение

В настоящее время не существует аналитических методов для точного определения характеристик сетей массового обслуживания, включающих в качестве узлов системы массового обслуживания (СМО) $G/G/1$ или $G/G/m$, и, как следствие, это отражается на степени адекватности стохастических сетевых моделей реальным компьютерным и телекоммуникационным сетям и на качестве принимаемых проектных решений. Рассмотрим открытую сетевую модель с матрицей вероятностей передач $P = \{p_{ij}\}$, ($i, j = 1, \dots, n$), где p_{ij} — вероятность того, что требование, покидающее узел S_i , поступит в узел S_j . В телекоммуникационных сетях аналогом этой матрицы будет "матрица тяготений узлов", состоящая в основном из 0 и 1. Далее рассмотрим полную структуру отдельного узла с номером i сетевой модели (рис. 1) [1].

Решением системы уравнений равновесия потоков относительно интенсивностей λ_i на входе и выходе каждого узла сети в установившемся режиме определяем средние значения интервалов времен между соседними требованиями $\bar{\tau}_i = \lambda_i^{-1}$ для каждого потока в сети:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^n p_{ji} \lambda_j, \quad (i = 1, \dots, n), \quad (1)$$

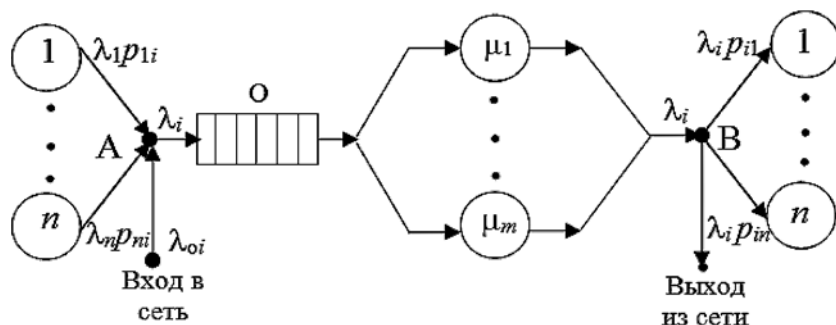


Рис. 1. Структура i -й СМО сети (А — точка композиции потоков, В — точка декомпозиции потоков)

где λ_{0i} — интенсивность потока извне в i -й узел (рис. 1). Система (1) — это уравнения баланса потоков на уровне первого момента распределения интервалов поступления требований в сеть массового обслуживания, позволяющие декомпозировать сеть на отдельные узлы. Уравнений (1) достаточно для декомпозиции марковской сети, но недостаточно для немарковской сети.

Постановка задачи

Для повышения точности расчетов сетей массового обслуживания требуются уравнения баланса (равновесия) потоков на уровне моментов высшего порядка распределения интервалов поступления требований. Это связано с непуассоновским характером потоков, циркулирующих в сети массового обслуживания. Вывод уравнений баланса сопряжен со многими проблемами. Рассмотрим некоторые из них.

На рис. 1 точка А — точка композиции (агрегирования) потоков, точка В — точка декомпозиции (вероятностного разрежения) агрегированного потока. Из рис. 1 и уравнений (1) следует, что на входе i -го узла агрегируются (мультиплицируются) (знак Σ в уравнениях (1)) разреженные исходящие потоки $(p_j \lambda_j)$ от других узлов. Таким образом, для вывода уравнений баланса на уровне моментов высшего порядка нужны будут: 1) математические модели мультиплицирования и вероятностного разрежения потоков на уровне необходимого числа моментов случайного интервала между требованиями; 2) математическая модель исходящего потока на уровне необходимого числа моментов интервалов между требованиями входящего потока, а также времени обслуживания в узле. Эти модели будут отправной точкой для вывода уравнений баланса потоков относительно моментов распределения интервалов для их последующего восстановления.

Решение задачи

Узел сети массового обслуживания представляет собой в общем случае многоканальную (с равным доступом) систему G/G/m с бесконечной очередью. Мы же ограничимся случаем одноканальной системы G/G/1 и рассмотрим отдельно следующие случаи: 1) коэффициенты вариаций интервала между поступлениями требований и времени обслуживания в узлах сети меньше 1 ($c_\lambda < 1$, $c_\mu < 1$); 2) коэффициенты вариаций интервала между поступлениями требований и времени обслуживания больше 1 ($c_\lambda \geq 1$, $c_\mu \geq 1$); 3) смешанный случай, когда один из коэффициентов вариаций больше 1, другой меньше 1.

В первом случае в качестве модели узла будем рассматривать систему M⁻/M⁻/1 с запаздыванием во времени [2]. В системе с запаздыванием M⁻/M⁻/1 на вход поступают требования, случайные интервалы между которыми распределены по закону

$$a(t) = \begin{cases} \lambda e^{-\lambda(t-t_0)}, & t \geq t_0; \\ 0, & 0 \leq t < t_0. \end{cases} \quad (2)$$

Аналогично распределено и время обслуживания:

$$b(t) = \begin{cases} \mu e^{-\mu(t-t_0)}, & t \geq t_0; \\ 0, & 0 \leq t < t_0. \end{cases} \quad (3)$$

Эта система впервые подробно описана в работе [2], где для нее представлено решение по среднему времени ожидания в очереди. Распределения (2) и (3), содержащие два параметра, позволяют аппроксимировать законы распределения на уровне двух первых моментов в случае $c_\lambda < 1$, $c_\mu < 1$.

Во втором случае в качестве модели узла будем рассматривать систему H₂/H₂/1 [3], которая описывается гиперэкспоненциальными входными распределениями 2-го порядка с функциями плотностей

$$a(t) = p\lambda_1 e^{-\lambda_1 t} + (1-p)\lambda_2 e^{-\lambda_2 t}; \quad (4)$$

$$b(t) = q\mu_1 e^{-\mu_1 t} + (1-q)\mu_2 e^{-\mu_2 t}. \quad (5)$$

Функции (4) и (5) содержат три неизвестных параметра и позволяют аппроксимировать законы распределения на уровне трех первых моментов в случае $c_\lambda \geq 1$, $c_\mu \geq 1$. Подробно система H₂/H₂/1 описана в работе [3], где для нее также представлено решение по среднему времени ожидания в очереди.

Система H₂/M/1 описывается гиперэкспоненциальным входным распределением 2-го порядка с функцией плотности (4) и функцией плотности времени обслуживания (3).

Расчеты характеристик систем M⁻/M⁻/1, H₂/H₂/1 и H₂/M/1 автоматизированы [4], и программа их расчета описана в работе [5]. Система M/H₂/1 относится к классу систем M/G/1 и легко может быть рассчитана с помощью формулы Полячека—Хинчина. Таким образом, автоматический расчет узловых характеристик сетей математически и программно обеспечен.

Третий случай можно рассматривать как комбинацию рассмотренных двух первых случаев. Для каждого узла сети задаем числовые характеристики случайного времени обслуживания: $(\bar{\tau}_{\mu i}, \tau_{\mu i}^2, \tau_{\mu i}^3)$ — два или три первых начальных момента времени обслуживания в зависимости

от рассмотренных выше случаев. Для внешнего потока задаем совокупность таких же моментов $(\bar{\tau}_{0i}, \tau_{0i}^2, \tau_{0i}^3)$ времени между соседними требованиями потока, входящего в узел S_i . Зная характеристики внешнего потока, решаем задачу восстановления двух или трех первых начальных моментов интервалов между требованиями во всех потоках такой сетевой модели в зависимости от рассматриваемого случая.

При выводе уравнений равновесия для определения дисперсий интервалов в потоках на основе доказанных утверждений [6], в работе получены следующие результаты.

Агрегирование (мультиплексирование) двух потоков

Функция распределения интервала времени τ_Σ результирующего потока при мультиплексировании двух потоков с интенсивностями λ_1 и λ_2 определяется интегральным соотношением

$$F_{\tau_\Sigma}(t) = 1 - \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \{ [1 - F_{\tau_1}(t)] \int_t^\infty [1 - F_{\tau_2}(u)] du + [1 - F_{\tau_2}(t)] \int_t^\infty [1 - F_{\tau_1}(u)] du \}. \quad (6)$$

Тогда начальные моменты интервала времени τ_R в результирующем потоке событий при агрегировании двух независимых потоков с интенсивностями $\lambda_1 = \bar{\tau}_1^{-1}$ и $\lambda_2 = \bar{\tau}_2^{-1}$ с функциями распределения вероятностей (ФРВ) $F_{\tau_1}(t)$ и $F_{\tau_2}(t)$ могут быть определены следующим образом [6]:

$$\bar{\tau}_R = 1/\lambda_\Sigma; \quad (7)$$

$$\bar{\tau}_R^2 = 2 \frac{\lambda_1 \lambda_2}{\lambda_\Sigma} \int_0^\infty g_1(t) g_2(t) dt; \quad (8)$$

$$\bar{\tau}_R^3 = 6 \frac{\lambda_1 \lambda_2}{\lambda_\Sigma} \int_0^\infty t g_1(t) g_2(t) dt, \quad (9)$$

где функции $g_j(t) = \int_t^\infty [1 - F_{\tau_j}(u)] du$, а значение $\lambda_\Sigma = \lambda_1 + \lambda_2$.

Здесь случайная величина τ_R не является суммой величин τ_1 и τ_2 , т. е. здесь не имеет место композиция законов распределений. Вместе с тем результирующий поток в этом случае не будет рекуррентным, кроме случая пуассоновских составляющих.

Из выражений (8) и (9) следует, что моменты интервалов между событиями результирующего потока, начиная со второго, нельзя выразить в элементарных функциях через моменты состав-

ляющих потока, кроме случая пуассоновских составляющих. Поэтому для практического применения выражения (6) исходные неизвестные ФРВ $F_{\tau_j}(t)$ интервалов аппроксимируем функцией смешенного экспоненциального распределения в случае, когда коэффициенты вариаций составляющих $c_{\lambda_j} \leq 1$ ($j = 1, 2$) [6]:

$$F_j(t) = \begin{cases} 0, & t \leq \tau_{j1}; \\ 1 - \exp\{-(t - \tau_{j1})/\tau_{j2}\}, & t \geq \tau_{j1}. \end{cases} \quad (10)$$

В случае $c_{\lambda_j} > 1$ исходные функции распределений аппроксимируем гиперэкспоненциальным распределением 2-го порядка:

$$F_j^*(t) = 1 - p_j \exp(-t/\tau_{j1}^*) - (1 - p_j) \exp(-t/\tau_{j2}^*). \quad (11)$$

В смешанном случае одну функцию аппроксимируем соотношением (10), а другую — (11). Тогда задача определения начальных моментов случайного интервала между событиями в результирующем потоке по выражениям (8) и (9) сводится к вычислению табличных интегралов.

Замечание. Если же вместо функций (10) и (11) рассматривать более общие законы распределений типа Вейбулла или гамма, задача определения начальных моментов усложняется в связи с появлением специальных функций в интегралах.

Задавая исходные агрегируемые потоки на уровне начальных моментов интервалов между событиями в потоках, неизвестные параметры ФРВ (10) и (11) находим известным методом моментов. При аппроксимации потоков на уровне двух первых моментов получим следующие значения неизвестных параметров (10) и (11) (1-й и 3-й случаи):

$$\begin{aligned} \tau_{j1} &= \bar{\tau}_j - \sqrt{\tau_j^2 - \bar{\tau}_j^2}; \quad \tau_{j1} = \sqrt{\tau_j^2 - \bar{\tau}_j^2}; \quad \tau_{j1}^* = \bar{\tau}_j / (2p_j); \\ \tau_{j2}^* &= \bar{\tau}_j / [2(1 - p_j)]; \quad p_j = 1/2 \pm \sqrt{1 - 2\bar{\tau}_j^2 / \tau_j^2}; \\ &(j = 1, 2). \end{aligned}$$

Аппроксимация потоков на уровне трех моментов потребует уже решения системы трех уравнений. Запишем уравнения для определения начальных моментов распределения (11):

$$\begin{aligned} p_j \tau_{j1}^* + (1 - p_j) \tau_{j2}^* &= \bar{\tau}_j; \\ 2p_j (\tau_{j1}^*)^2 + 2(1 - p_j) (\tau_{j2}^*)^2 &= \bar{\tau}_j^2; \\ 6p_j (\tau_{j1}^*)^3 + 6(1 - p_j) (\tau_{j2}^*)^3 &= \bar{\tau}_j^3, \end{aligned} \quad (12)$$

решив которые, найдем искомые параметры распределения (11). Как видно из системы (12), в этом случае задача определения параметров распреде-

ления (11) усложняется, но и решение этой задачи автоматизировано и описано в работах [4, 5].

Замечание. Для существования решения системы (12) необходимо и достаточно выполнение условия [7]: $\overline{\tau_j^3 \tau_j} \geq 1,5 \overline{\tau_j^2}$. Следовательно, при аппроксимации произвольных законов распределения гиперэкспоненциальным законом необходимо придерживаться данного ограничения.

Разрежение (демультиплексирование) потоков

При вероятностном разрежении произвольно-го рекуррентного потока со средним значением $\bar{\tau}$ и дисперсией D_τ времени между событиями (точка B на рис. 2), в которой требования с вероятностью $p \neq 0$ уходят из потока (просеянный поток 2), среднее значение и дисперсия времени между соседними событиями в разреженном потоке 2 равны [6]:

$$\bar{\tau}_p = \bar{\tau}/p; \quad (13)$$

$$D_{\tau_p} = D_\tau/p + \bar{\tau}^2 q/p^2, \quad (14)$$

где $q = 1 - p$.

Выражения (13) и (14) получены следующим образом. Представим поток событий с параметрами $(\bar{\tau}, D_\tau)$, где $\bar{\tau}, D_\tau$ — среднее значение и дисперсия времени между соседними событиями, как последовательность случайных точек $\{t_i\}$ на оси времени $0t$. Случайные интервалы времени между ними обозначим соответственно $\tau_1, \tau_2, \tau_3, \dots$. При разрежении потока 1 случайный интервал времени τ_p в потоке 2 равен: τ_1 — с вероятностью p ; $(\tau_1 + \tau_2)$ — с вероятностью pq ; $(\tau_1 + \tau_2 + \tau_3)$ — с вероятностью pq^2 и т. д. Запишем выражение для величины τ_p следующим образом:

$$\tau_p = \begin{cases} \tau_1 & \text{с вероятностью } p; \\ (\tau_1 + \tau_2) & \text{с вероятностью } pq; \\ (\tau_1 + \tau_2 + \tau_3) & \text{с вероятностью } pq^2; \\ \dots & \dots \\ (\tau_1 + \tau_2 + \dots + \tau_i) & \text{с вероятностью } pq^{i-1}; \\ \dots & \dots \end{cases} \quad (15)$$

Определим математическое ожидание величины τ_p с учетом того факта, что поток рекуррентный, т. е. $M(\tau_i) = M(\tau)$, $M(\tau_i^2) = M(\tau^2)$:

$$\begin{aligned} M(\tau_p) &= M(\tau) p \sum_{i=1}^{\infty} i q^{i-1} = p M(\tau) (q + q^2 + q^3 + \dots)'_q = \\ &= p M(\tau) \left(\frac{q}{1-q} \right)'_q = p M(\tau) \frac{1}{(1-q)^2} = M(\tau)/p. \end{aligned}$$

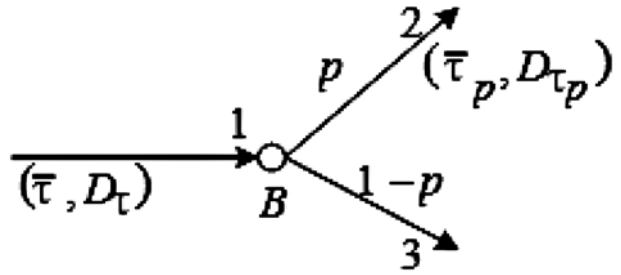


Рис. 2. Демультиплексирование потока (p — преобразование потока)

Для определения дисперсии D_{τ_p} вычислим второй начальный момент $M(\tau_p^2)$:

$$\begin{aligned} M(\tau_p^2) &= p \sum_{k=1}^{\infty} [k M(\tau^2) + (k^2 - k)(M(\tau))^2] q^{k-1} = \\ &= p \sum_{k=1}^{\infty} [k D_\tau + k^2 (M(\tau))^2] q^{k-1} = \\ &= p D_\tau \sum_{k=1}^{\infty} k q^{k-1} + p (M(\tau))^2 \sum_{k=1}^{\infty} k^2 q^{k-1} = \\ &= D_\tau/p + (1+q)(M(\tau))^2/p^2. \end{aligned}$$

Из полученных равенств для $M(\tau^2)$ и $M(\tau_p)$ следует справедливость выражений (13) и (14). Таким образом, можно определить при необходимости и третий начальный момент интервала в разреженном потоке, хотя это потребует больших усилий по сравнению с вычислением второго начального момента.

Теперь, после того как определены математические операции мультиплексирования и демультиплексирования потоков, по аналогии с уравнениями равновесия потоков на уровне их средних значений (1) можем записать уравнения равновесия относительно их дисперсий. Для этого повторно заметим, что на входе в i -й узел в общем случае агрегируются (мультиплексируются) разреженные (демультиплексированные) выходные потоки Π_{ij} от узлов j ($j = 0, 1, 2, \dots, n$). Дисперсии времен между событиями этих потоков, полученные по формуле (14), равны

$$D_{\Pi_{ji}} = \frac{1}{p_{ji}} \left(D_{\text{вых } j} + \frac{1-p_{ji}}{p_{ji} \lambda_j^2} \right), \quad i, j = 1, 2, \dots, n. \quad (16)$$

Тогда уравнения равновесия потоков на уровне дисперсий времен между событиями на входе и выходе i -го узла сетевой модели можно записать в виде уравнений

$$\begin{aligned} D_{\text{вх } i} &= \\ &= D(\Pi_{0i} * (\Pi_{1\text{вх},i} * \dots * (\Pi_{n-1\text{вх},i} * \Pi_{n\text{вх},i}))). \end{aligned} \quad (17)$$

Здесь и в дальнейшем, выражение $D(\Pi_{j-1, \text{вых}, i} * \Pi_{j, \text{вых}, i})$ означает операцию вычисления дисперсии попарно мультиплексируемых по формулам (7)—(8) выходных потоков от $(j-1)$ -го $(\Pi_{j-1, i})$ и j -го (Π_{ji}) узлов, поступающих на вход i -го узла. В выражении (17) дисперсию интервала между событиями в потоке Π_{0i} , поступающего на вход i -го узла от внешнего источника, обозначим D_{0i} .

Определение характеристик выходного потока

Числовые характеристики выходного потока из СМО могут быть определены по следующим выражениям [6, 12]:

$$\bar{\tau}_{\text{вых}} = \bar{\tau}_{\mu} + p'_0 \bar{\tau}'_{\lambda}; \quad (18)$$

$$D_{\text{вых}} = D_{\mu} + p'_0 D'_{\lambda} + p'_0(1 - p'_0)(\bar{\tau}'_{\lambda})^2, \quad (19)$$

где p'_0 — вероятность того, что обслуженное требование оставляет СМО пустой $\bar{\tau}'_{\lambda}$ и D'_{λ} — среднее значение и дисперсия остаточного времени τ'_{λ} , в течение которого СМО ожидает поступления непосредственно следующего требования, т. е. времени простоя СМО. Вывод выражений (18) и (19) основан на следующих фактах. Рассмотрим моменты времени t_1 ухода очередного требования из СМО и t_2 — ухода непосредственно следующего требования. Случайная величина $\Delta = t_2 - t_1$ существенно зависит от состояния СМО в момент ухода очередного требования. Если в момент времени t_1 СМО окажется занятой, то величина Δ будет равна времени обслуживания τ_{μ} непосредственно следующего требования. Если же в момент времени t_1 СМО окажется пустой, то величина Δ будет равна сумме времени обслуживания τ_{μ} и остаточного времени τ'_{λ} . Тогда можем записать следующие выражения для случайной величины Δ и ее квадрата Δ^2 по аналогии с законом распределения вероятностей:

$$\Delta = \begin{cases} \tau_{\mu} & \text{с вероятностью } 1 - p'_0; \\ (\tau_{\mu} + \tau'_{\lambda}) & \text{с вероятностью } p'_0; \end{cases} \quad (20)$$

$$\Delta^2 = \begin{cases} \tau_{\mu}^2 & \text{с вероятностью } 1 - p'_0; \\ (\tau_{\mu} + \tau'_{\lambda})^2 & \text{с вероятностью } p'_0. \end{cases}$$

Отсюда, переходя к математическому ожиданию и дисперсии величины Δ и учитывая, что $M(\tau_{\text{вых}}) = M(\Delta)$ и $D_{\text{вых}} = M(\Delta^2) - [M(\Delta)]^2$, после преобразований получим формулы (18) и (19).

Заметим, что при необходимости числовые характеристики выходного потока также можно определить включительно до третьего момента.

Для рассмотренных выше СМО $M^-/M^-/1$, $H_2/H_2/1$ и $H_2/M/1$ при выводе решения для сред-

него времени ожидания на основе метода спектрального разложения решения интегрального уравнения Линдли автоматически определяется константа K — вероятность того, что поступающее в систему требование застаёт ее свободной [2, 3]. Поэтому для таких узлов сети массового обслуживания в качестве вероятности p'_0 в выражениях (18)—(20) можно будет использовать соответствующее значение константы K , и расчет характеристик выходного потока упростится.

Таким образом, все основные проблемы для приближенной декомпозиции сети массового обслуживания на отдельные узлы разрешены, и для этого можно использовать приведенные выше расчетные выражения. Система (17) совместно с (16) и представляют собой уравнения равновесия потоков в сети МО на уровне дисперсий интервалов времени для их восстановления. Решение уравнений (1) и (17) совместно с выражениями для дисперсий выходных потоков (19) является основой метода баланса потоков, позволяющего декомпозировать сети МО на уровне двух первых моментов распределений вероятностей, т. е. определить средние значения и дисперсии интервалов времени между требованиями в потоках, циркулирующих в сети в условиях стационарного режима.

Решение уравнений равновесия

Уравнения (1), (17) совместно с (16) и (19) образуют уравнения равновесия относительно средних значений и дисперсий распределений интервалов времени между соседними требованиями в потоках сети. Это вынужденный подход к анализу потоков на уровне двух первых моментов распределений интервалов путем аппроксимации произвольных законов распределений в условиях неполной информации о них. Другого подхода к анализу немарковских сетей массового обслуживания, к сожалению, в настоящее время не существует.

В стационарном случае система уравнений (1) имеет единственное решение, так как ее определитель не равен нулю, и, как отмечено в работе [1], эта система представляет одну из форм закона сохранения.

Система (17) совместно с (16) решается методом итераций. На каждой итерации дисперсия $D_{j, \text{вых}}$ времени между соседними заявками в выходном из j -й СМО потоке определяется методом двумерной диффузионной аппроксимации СМО, тем самым уточняются значения дисперсий $D_{i, \text{вх}}$ входных потоков [12]. В программе процесс уточнения дисперсий завершается по достижении 1 % относительной погрешности. При решении мно-

гих задач из области сетевых технологий, а именно: при определении загрузки каналов или других ресурсов сети, а также при определении задержек "источник—адресат" в сетях связи, в матрице вероятностей передач $P = \{p_{ij}\}(i, j = 1, \dots, n)$ элементы p_{ij} принимают значения в основном 0 и 1. В связи с этим уравнения (1) и (17) в таком случае упрощаются.

Приведенные выше формулы по агрегированию и вероятностному разрежению потоков в работе [6] проверены с помощью имитационного моделирования по специально разработанным программам агрегирования и разрежения потоков. Для этого проводили серии из пяти испытаний, в каждом из которых генерировали по 10 000 интервалов. Интервалы генерировали по экспоненциальному, равномерному, нормальному, Вейбулла, гамма и другим законам распределений времени между требованиями. Результаты имитационного моделирования полностью подтверждают справедливость утверждений. Относительная погрешность моделирования при этом не превышает 5 %.

Пример расчета

В подтверждение приведенных выше расчетных формул проведен расчет сетевой модели из трех узлов. Исходные данные приведены на рис. 3, а результаты расчетов — на рис. 4.

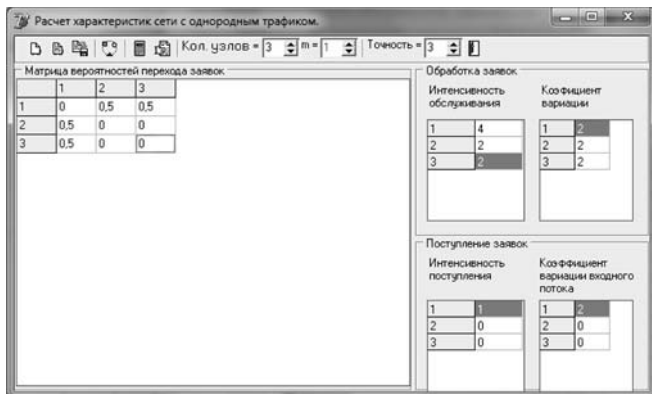


Рис. 3. Исходные данные к расчетному примеру

Узловые характеристики:

Характеристики/Узлы	1	2	3
Интенсивность потока	2,0	1,0	1,0
Загрузка	0,500	0,500	0,500
Ср. время ожидания	0,920	1,560	1,560
Ср. число заявок	2,339	2,060	2,060

Сетевые характеристики:

Ср. время ожидания	Ср. задержка (время отклика)	Ср. длина очереди
4,958	6,458	4,958

Время расчета: 0,265с

Рис. 4. Результаты расчетов трехузловой модели

Заключение

Совместно предложенные модели математического мультиплексирования и демультиплексирования потоков, а также выражения для определения числовых характеристик выходного потока позволяют записать уравнения равновесия относительно средних значений и дисперсий распределений интервалов времени между соседними требованиями в сетях массового обслуживания при произвольных законах поступления и обслуживания.

При общих допущениях о вероятностных распределениях времени между соседними требованиями в входных потоках и времени обслуживания в узлах предложенный подход позволяет определить средние значения и дисперсии распределения интервалов времени между требованиями всех потоков сети массового обслуживания, а также все основные показатели функционирования таких сетей. Такой подход можно расширить при необходимости и для моментов 3-го порядка, что с точки зрения теории вероятностей точнее, чем расчет на уровне двух моментов распределений.

В условиях неполной информации о законах распределений потоков предложенный подход к анализу производительности немарковских сетей массового обслуживания является на данный момент приемлемым.

Список литературы

1. Клейнрок Л. Теория массового обслуживания: Пер. с англ. / Под ред. В. И. Неймана. М.: Машиностроение, 1979. 432 с.
2. Тарасов В. Н., Бахарева Н. Ф., Блатов И. А. Анализ и расчет системы массового обслуживания с запаздыванием // Автоматика и телемеханика. 2015. № 11. С. 51—59.
3. Тарасов В. Н. Исследование систем массового обслуживания с гиперэкспоненциальными входными распределениями // Проблемы передачи информации. 2016. № 1. С. 16—26.
4. Тарасов В. Н., Бахарева Н. Ф., Липилина Л. В. Автоматизация расчета характеристик систем массового обслуживания для широкого диапазона изменения их параметров // Информационные технологии. 2016. № 12. С. 952—957.
5. Тарасов В. Н., Липилина Л. В. Программный комплекс расчета характеристик систем массового обслуживания типа $H_2/H_2/1$, $H_2/M/1$ и $M/M/1$ с запаздыванием во времени. Св-во о гос. регистрации программы для ЭВМ № 2016612657 от 03.03.2016.
6. Бахарева Н. Ф., Тарасов В. Н. Аппроксимативные методы и модели массового обслуживания. Исследование компьютерных сетей. Самара: Изд-во СНЦ РАН, 2011. 327 с.
7. Whitt W. Approximating a point process by a renewal process, I: two basic methods // Operation Research. 1982. No. 1. P. 125—147.
8. Тарасов В. Н., Бахарева Н. Ф., Горелов Г. А., Малахов С. В. Анализ входящего трафика на уровне трех моментов распределений временных интервалов // Информационные технологии. 2014. № 9. С. 54—59.

9. **Тарасов В. Н., Каргашевский И. В.** Определение среднего времени ожидания требований в управляемой системе массового обслуживания $H_2/H_2/1$ // Системы управления и информационные технологии. 2014. № 3. С. 92–96.

10. **Myaskja A.** An improved heuristic approximation for the GI/GI/1 queue with bursty arrivals // *Teletraffic and datatraffic in a Period of Change, ITC-13*. Elsevier Science Publishers. 1991. P. 683–688.

11. **Тарасов В. Н., Бахарева Н. Ф., Липилина Л. В.** Математическая модель телетрафика на основе системы G/M/1 и

результаты вычислительных экспериментов // Информационные технологии. 2016. № 2. С. 121–126.

12. **Кругликов В. К., Тарасов В. Н.** Анализ и расчет сетей массового обслуживания методом двумерной диффузионной аппроксимации // Автоматика и телемеханика. 1983. № 8. С. 74–83.

13. **URL:** <https://tools.ietf.org/html/rfc3393>. RFC 3393 IP Packet Delay Variation Metric for IP Performance Metrics (IPPM) (дата обращения: 26.02.2016).

V. N. Tarasov, D. Sc., Professor, Head of Chair, e-mail: veniamin_tarasov@mal.ru,
N. F. Bakhareva, D. Sc., Professor, Head of Chair, e-mail: nadin1956_04@inbox.ru,
L. V. Lipilina, Postgraduate Student, e-mail: mila199113@gmail.com,
Povolzhsky State University of Telecommunications and Informatics, Samara, 443010, Russia

Analysis of Non-Markov Queuing Networks on the Basis of the Equations Balance of Flows

The article proposes models of mathematical multiplexing and demultiplexing of flows, as well as expressions for determining the numerical characteristics of the distribution of the outflow intervals from the queuing system (QS). Together, they allow us to write the equilibrium equations for the mean and variance of the distribution of time intervals between consecutive requirements in queuing networks with arbitrary laws receipts and services. On the place of nodes of a queuing network are considered QS M/M/1 with delay in time, $H_2/H_2/1$ and $H_2/M/1$. Under general assumptions about the probability distribution of time intervals between requirements in the input flows and the time of service in the nodes, the proposed approach makes it possible to determine the average values and variances of time intervals between the requirements of all queuing network flows, as well as all the main indicators of the functioning of such networks. Such an approach can also be extended, if necessary, for third-order moments, which from the point of view of probability theory is more accurate than the calculation at the level of two moments of the distributions. Such an approach can also be extended, if necessary, for third-order moments. From the point of view of probability theory, this will be more accurate than calculating at the level of two moments of distribution. Under incomplete information on the laws of flows distributions, the proposed approach to analyzing the performance of non-Markov queuing networks is currently acceptable.

Keywords: queuing networks, queuing systems $M/M/1$ with delay in time, $H_2/H_2/1$ and $H_2/M/1$, aggregation and rarefaction of flows

References

1. **Kleinrock L.** *Teoriya massovogo obsluzhivaniya*, Moscow, Mashinostroenie, 1979, 432 p. (in Russian).

2. **Tarasov V. N., Bakhareva N. F., Blatov I. A.** *Analiz i raschet sistemy massovogo obsluzhivaniya s sapasdyvaniem* (Analysis and Calculation of Queuing System with Delay), *Avtomatica i Telemekhanika*, 2015, no. 11, pp. 51–59 (in Russian).

3. **Tarasov V. N.** *Issledovanie sistem massovogo obsluzhivaniya s gipereksponencial'nymi vhodnymi raspredeleniyami* (Analysis of queues with hyperexponential arrival distributions), *Problemy Peredachi Informacii*, 2016, no. 1, pp. 16–26 (in Russian).

4. **Tarasov V. N., Bakhareva N. F., Lipilina L. V.** *Avtomatizatsiya rascheta harakteristik sistem massovogo obsluzhivaniya dlya shirokogo diapazona izmeneniya ih parametrov* (Automation for calculating characteristics queuing system for a wide range changing their parameters), *Informacionnye Tehnologii*, 2016, no. 12, pp. 952–957 (in Russian).

5. **Tarasov V. N., Lipilina L. V.** *Programmnyi complex rascheta harakteristik sistem massovogo obsluzhivaniya tipa $H_2/H_2/1$, $H_2/M/1$ i $M/M/1$ s sapasduvaniem vo vremeni* (Program complex of calculation of characteristics of queuing systems like $H_2/H_2/1$, $H_2/M/1$ and $M/M/1$ with delay in time), Certificate on the state registration of the computer program no. 2016612657 from 03.03.2016 (in Russian).

6. **Bakhareva N. F., Tarasov V. N.** *Approksimativnye metody i modeli massovogo obsluzhivaniya Issledovanie komp'yuternykh setej* (Approximative methods and queuing models Investigation of computer networks), Samara, Publishing house of SNC RAN, 2011, 327 p. (in Russian).

7. **Whitt W.** Approximating a point process by a renewal process, I: two basic methods, *Operation Research*, 1982, no. 1, pp. 125–147.

8. **Tarasov V. N., Bakhareva N. F., Gorelov G. A., Malahov S. V.** *Analiz vkhodyashchego trafika na urovne trekh momentov raspredeleniy vremennykh intervalov* (Analyzing the Incoming Traffic at the Three Moments Distribution of Time Intervals), *Informacionnye Tehnologii*, 2014, no. 9, pp. 54–59 (in Russian).

9. **Tarasov V. N., Kartashevsky I. V.** *Opreделение srednego vremeni ogidaniya trebovaniy v upravlyemoy sisteme massovogo obsluzhivaniya $H_2/H_2/1$* (Determination of the average waiting time requirements in a controlled queuing system), *Sistemy Upravleyiya i Informacionnye Tehnologii*, 2014, no. 3, pp. 92–96 (in Russian).

10. **Myaskja A.** An improved heuristic approximation for the GI/GI/1 queue with bursty arrivals, *Teletraffic and datatraffic in a Period of Change, ITC-13*, Elsevier Science Publishers, 1991, pp. 683–688.

11. **Tarasov V. N., Bakhareva N. F., Lipilina L. V.** *Matematicheskaya model' teletrafika na osnove sistemy G/M/1 i rezul'taty vychislitel'nykh ehksperimentov* (Mathematical Model of Teletraffic on the Based G/M/1 System and Results of Computational Experiment), *Informacionnye Tehnologii*, 2016, no. 2, pp. 121–126 (in Russian).

12. **Kruglikov V. K., Tarasov V. N.** *Analiz i raschet setei massovogo obsluzhivaniya metodom dvumernoi diffuzionnoi approksimacii* (Analysis and calculation of queuing networks by method of two-dimensional diffusion approximation), *Avtomatica i Telemekhanika*, 1983, no. 8, pp. 74–83 (in Russian).

13. **Available** at: <https://tools.ietf.org/html/rfc3393>. RFC 3393 IP Packet Delay Variation Metric for IP Performance Metrics (IPPM) (date of access: 26.02.2016).

В. А. Стенников, чл.-корр., директор, e-mail: sva@isem.irk.ru,
Е. А. Барахтенко, канд. техн. наук, ст. науч. сотр., e-mail: barakhtenko@isem.irk.ru,
Д. В. Соколов, канд. техн. наук, ст. науч. сотр., e-mail: sokolov_dv@isem.irk.ru,
Федеральное государственное бюджетное учреждение науки
Институт систем энергетики им. Л. А. Мелентьева
Сибирского отделения Российской академии наук (ИСЭМ СО РАН), г. Иркутск

Разработка принципов построения интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики*

Представлен методический подход к построению интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики различного типа. Автоматизированное построение интегрированной графической среды выполняется на основе компьютерной модели трубопроводной системы и онтологий на базе концепции Model-Driven Engineering и рефлексивного программирования.

Ключевые слова: интегрированная графическая среда, методический подход, автоматизация программирования, Model-Driven Engineering, метапрограммирование, рефлексивное программирование, онтология, трубопроводная система, энергетика

Введение

Современные трубопроводные системы энергетики (ТПС) (тепло-, водо-, нефте-, газоснабжения и др.) представляют собой уникальные по масштабам и сложности инженерные сооружения, обладающие все возрастающим значением в энергетике, экономике, промышленности, коммунально-бытовом секторе и других сферах жизни страны и общества. Эффективное решение задач их проектирования, реконструкции, эксплуатации и диспетчерского управления невозможно без применения интегрированной графической среды. Эта среда представляет собой программную систему, позволяющую решать информационные (работа с компьютерной моделью ТПС и ее элементами на плане местности в интерактивном режиме), расчетные (решение инженерных задач) и аналитические (просмотр графиков, таблиц и результатов расчетов) задачи в рамках единого интерфейса пользователя. Современная среда должна обеспечивать работу в режиме многоуровневого моделирования, когда моделируемая ТПС может состоять из множества расчетных схем различных типов (схемы сетей, насосных станций, источников и т. д.), и элемент ТПС (например, источник) может быть представлен как элементом расчетной схемы, так и собственной расчетной схемой.

*Исследования выполняются при финансовой поддержке Российского фонда фундаментальных исследований (грант № 16-07-00948).

Общность структурно-топологических свойств ТПС позволяет ставить задачу разработки единой программной системы, являющейся интегрированной графической средой для компьютерного моделирования ТПС различного типа. В проектных и эксплуатационных организациях применяется специализированное программное обеспечение, ориентированное на конкретный тип ТПС и определенные классы решаемых задач. В настоящее время отсутствует описание методических подходов к построению интегрированных графических сред, обладающих универсальностью и возможностью моделирования различных типов ТПС в рамках единой программной системы.

В данной статье предложен методический подход к автоматизированному построению интегрированной графической среды для компьютерного моделирования ТПС. На базе этого подхода реализована программная система, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС различного типа.

Описание проблемы и постановка задач исследования

Компьютерное моделирование ТПС начинается с построения ее модели, которая представляет собой совокупность структур данных, задающих конфигурацию ТПС, описывает состав ее оборудования и его характеристики, состояние элементов и их свойства (технические характеристики, гидравлические параметры и граничные условия).

ТПС различного типа обладают общностью структурно-топологических свойств и физических законов течения транспортируемой среды [1], что позволяет сформулировать следующие общие положения, характерные для методологии их компьютерного и математического моделирования:

- все ТПС могут быть смоделированы графом, вершины которого соответствуют узлам (источники, узлы-соединения, потребители), а дуги — ветвям (трубопроводы, активные ветви с насосными станциями, регуляторами давления или расхода);
- задачи математического моделирования ТПС обладают общностью своих содержательных и математических постановок, а применяемые для их решения методы, алгоритмы и специализированное программное обеспечение могут носить универсальный характер (не зависеть от типа ТПС);
- компьютерная модель ТПС определенного типа может быть представлена как совокупность графа, описывающего конфигурацию этой системы, и набора графических и математических моделей, описывающих свойства ее элементов.

Сформулированные положения показывают, что существует возможность создания интегрированной графической среды в виде единой универсальной программной системы, которая обеспечит возможность работы пользователя с активными графическими моделями ТПС в интерактивном режиме, когда программная система динамически настраивается на конкретный тип ТПС и состав ее оборудования путем интеграции соответствующих моделей элементов ТПС.

Авторы рассматривали возможность применения объектно-ориентированного и компонентного подходов для построения единой программной системы, позволяющей моделировать различные типы ТПС. Практическое применение этих подходов приводит к возникновению сложной проблемы, вызванной необходимостью интеграции в программную систему большого числа программных компонентов, используемых для моделирования ТПС различных типов, элементы которых могут быть представлены широким набором оборудования и могут находиться в различных состояниях.

Высокая трудность поддержания в актуальном состоянии всего набора необходимых программных компонентов позволяет сделать вывод, что для реализации интегрированной графической среды необходимо привлечение парадигм программирования, ориентированных на автоматизацию этапов построения программного обеспечения (ПО).

ТПС определенного типа обладают характерными для этого типа свойствами, составом

прикладных задач и используемым для их решения специализированным ПО. Знания о них необходимо сохранить в форме, пригодной для обработки программными системами, и использовать при решении различных задач. Для этого необходимо организовать хранение этих знаний в виде онтологий [2], которые можно использовать для автоматизации этапов построения ПО, информационного наполнения пользовательского интерфейса и обеспечения эффективной работы с компьютерной моделью ТПС.

Методический подход и его особенности

В результате проведенных авторами исследований предложен методический подход, в рамках которого объединены концепция Model-Driven Engineering ("управляемая моделями разработка", MDE) и рефлексивное программирование.

Концепция MDE представляет собой совокупность методических подходов к автоматизированному построению сложных программных систем на основе предварительно разработанных моделей [3—5]. В рамках предлагаемого подхода MDE адаптирована к особенностям предметной области компьютерного моделирования ТПС, а автоматизированное построение программной системы выполняется на основе компьютерной модели ТПС и онтологий.

Рефлексивное программирование является одним из видов метапрограммирования и представляет собой расширение парадигмы объектно-ориентированного программирования [6—7]. Рефлексивное программирование позволяет выполнять операции, которые в рамках объектно-ориентированного программирования выполнить невозможно, среди которых наиболее важными являются следующие: изучение классов во время выполнения программы, определение их полей, методов и конструкторов; создание новых экземпляров объектов по названию класса с использованием конструкторов; установка и получение значений полей по их названию; вызов методов по их названию и описанию аргументов; гибкая работа с массивами и контейнерами (коллекциями). Рефлексивное программирование позволяет отслеживать структуру программной системы и динамически изменять состав программных компонентов во время ее работы.

Разработанный авторами методический подход включает:

- принципы автоматизированного построения интегрированной графической среды для компьютерного моделирования ТПС;
- архитектуру интегрированной графической среды, описывающую устройство программ-

ной системы, которая создается в автоматизированном режиме;

- методику автоматизированного построения программной системы на основе концепции MDE, рефлексивного программирования, онтологий и компьютерной модели ТПС.

Основные принципы, лежащие в основе методического подхода:

- программная система, ориентированная на моделирование конкретной ТПС, строится в автоматизированном режиме в контексте решения задачи построения компьютерной модели этой ТПС (рис. 1);
- автоматизированное построение программной системы и ее адаптация к конкретным особенностям моделируемой ТПС выполняется средствами рефлексивного программирования;

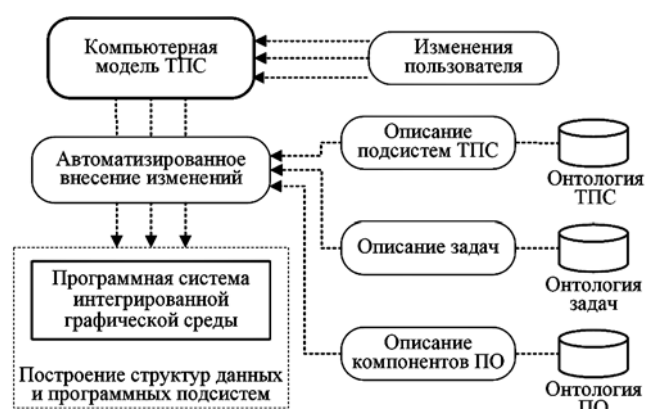


Рис. 1. Иллюстрация принципа построения интегрированной графической среды

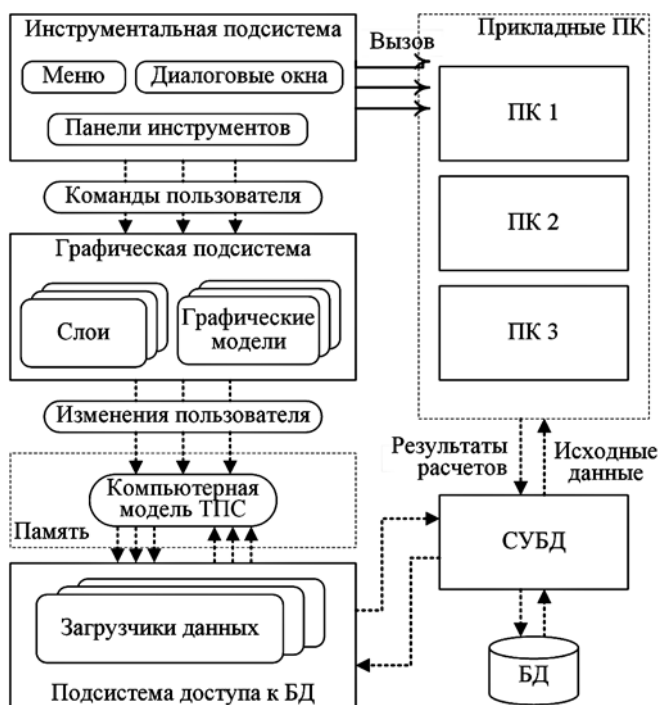


Рис. 2. Архитектура интегрированной графической среды

- для автоматизации этапов построения программной системы и информационного наполнения пользовательского интерфейса используются знания, хранение которых организовано в виде онтологий;
- знания, описывающие общие для всех типов ТПС свойства, задачи и используемое для их решения ПО, сохраняются в виде онтологий, а конкретные особенности моделируемой ТПС описываются ее компьютерной моделью. Авторами предложена архитектура интегрированной графической среды (рис. 2), которая включает следующие составляющие:

1) *графическую подсистему*, которая обеспечивает работу пользователя с активной графической моделью ТПС на плане местности и позволяет ему просматривать данные в удобном для восприятия виде и вносить необходимые изменения;

2) *инструментальную подсистему*, которая обеспечивает решение следующих задач: создает необходимый графический вид меню, позволяя пользователю решать задачи, соответствующие типу моделируемой ТПС; создает и предоставляет пользователю панели инструментов для рисования, редактирования, изменения состояний элементов активной модели ТПС; формирует диалоговые окна для настройки отображения модели ТПС, занесения необходимой информации и параметров, влияющих на решение прикладных задач; создает панели инструментов для рисования элементов плана местности и городской застройки;

3) *подсистему доступа к базам данных*, которая обеспечивает организацию работы с различными базами данных (БД), применяемыми для хранения и многократного использования компьютерных моделей ТПС, исходных данных и результатов расчетов, данных об объектах городской застройки.

Представленная архитектура является описанием устройства программной системы, которая создается в автоматизированном режиме и реализует интегрированную графическую среду для компьютерного моделирования ТПС.

При реализации интегрированной графической среды в соответствии с предлагаемым методическим подходом авторы ориентируются только на свободное ПО. В качестве базового языка программирования используется Java, выбор которого обусловлен поддержкой современных технологий объектно-ориентированного, компонентного и функционального программирования, а также встроенной поддержкой рефлексивного программирования и других видов метапрограммирования [6, 8]. Для работы с БД используется СУБД Firebird, которая обладает возможностью многопользовательского доступа и переносимостью между различными операционными системами.

Принципы построения онтологий

В рамках интегрированной графической среды используются три онтологии: онтология ТПС, онтология задач и онтология ПО. Для долговременного хранения онтологий авторы отдали предпочтение формату JSON¹, который является универсальным средством обмена данными между программными системами и предоставляет возможность организации хранения сложных (иерархических, сетевых) структур данных, а в современных платформах программирования существуют библиотеки, предназначенные для работы с этим форматом.

Загрузка содержимого онтологий в оперативную память выполняется загрузчиком онтологий (рис. 3), который по типу моделируемой ТПС считывает из файлов JSON соответствующее этому типу содержимое. Для выполнения этой операции используется библиотека GSON². Полученное описание записывается в структуры данных, реализующие представление содержимого онтологий в памяти.

Построение интегрированной графической среды для работы с ТПС конкретного типа осуществляется в автоматизированном режиме на основе описания этого типа систем из онтологии ТПС. Онтология ТПС включает следующие составляющие: описание иерархической структуры построения ТПС различного типа и назначения, классификацию используемого оборудования сетей и его технические характеристики, описание элементов ТПС и их свойств (технических характеристик, гидравлических параметров и граничных условий), классификацию и описание используемых математических моделей (например, уравнения, описывающие законы падения напора на ветвях сети, формулы расчета сопротивлений и др.).

На рис. 4 представлен фрагмент онтологии ТПС. ТПС состоит из элементов, которые подразделяются на узлы и ветви и могут находиться в одном из следующих физических состояний: рабочем, аварийном, отключенном или на реконструкции. Узлы,

в свою очередь, подразделяются на источники, узлы-соединения и потребители, а ветви — на активные и пассивные. Активные ветви содержат оборудование, оказывающее активное влияние на потокораспределение в сети. К ветвям этого типа относятся насосные станции и ветви с регуляторами расхода или давления. Пассивные ветви содержат трубопроводные участки ТПС, которые обладают такими свойствами, как длина и тип прокладки трубопровода. Каждый трубопроводный участок имеет конкретный тип трубопровода, характеризующийся диаметром и материалом. Для трубопровода имеется математическая модель, описывающая законы падения давления, скорость течения теплоносителя и выражение для расчета гидравлического сопротивления, а также графическая модель, определяющая способ отображения.

Онтология задач используется для настройки компьютерной модели ТПС на конкретную задачу и автоматизации процесса подготовки данных для вызова интегрированной графической средой прикладного ПО. Эта онтология содержит описание прикладных задач (например, оптимизации параметров многоконтурной сети) и методов их решения, описание алгоритмов, перечисление параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи.

Онтология ПО предназначена для хранения знаний, необходимых для автоматизации по-

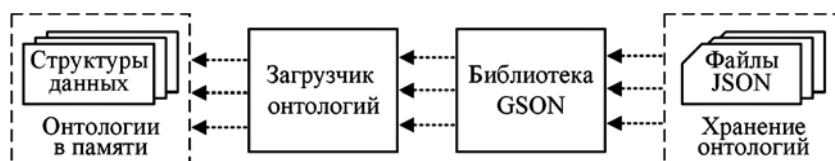


Рис. 3. Иллюстрация процесса загрузки содержимого онтологий

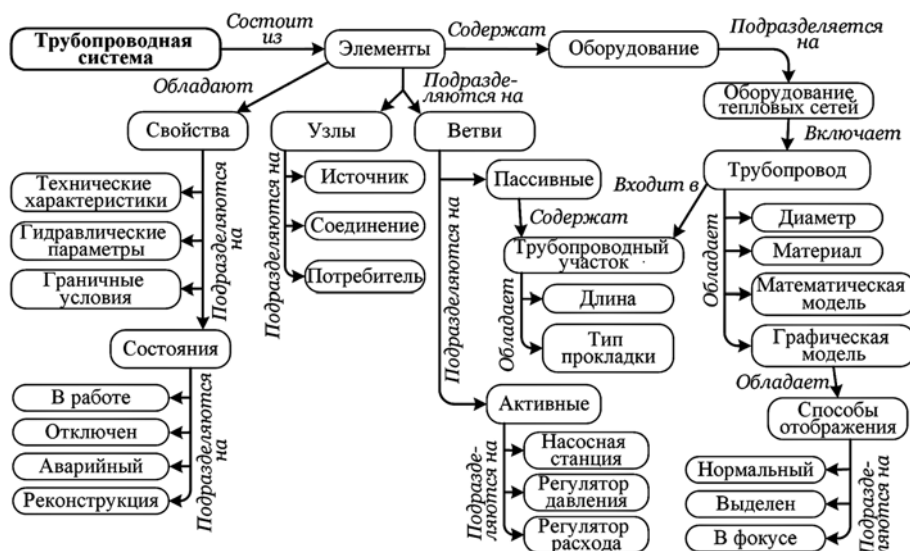


Рис. 4. Фрагмент онтологии трубопроводных систем

¹ <http://json.org/>

² <https://github.com/google/gson>

строения и использования ПО. Эта онтология содержит описание:

- программных комплексов, предназначенных для решения прикладных задач;
- программных компонентов, используемых для построения интегрированной графической среды;
- метаданных (входные и выходные параметры, описание форматов данных);
- технологий и интерфейсов доступа к программным компонентам.

Онтология ПО содержит описание следующих типов программных компонентов:

- компоненты инструментальной подсистемы, используемые для построения элементов пользовательского интерфейса (меню, панели инструментов, диалоговые окна и др.);
- компоненты, используемые для представления отдельных элементов (узлов и ветвей) графа моделируемой ТПС;
- компоненты, реализующие графические модели элементов ТПС определенного типа и в определенном рабочем состоянии;
- компоненты, реализующие структуры данных для описания отдельных элементов конкретных ТПС (технические характеристики, гидравлические параметры и граничные условия);
- компоненты графической подсистемы, используемые для отображения геометрических фигур, формирующих объекты городской застройки на плане местности (линии, полигоны, надписи и др.);
- компоненты доступа к БД, используемые для чтения и записи данных в соответствии с типом ТПС.

Методика автоматизированного построения

В соответствии с предложенной методикой автоматизированное построение интегрированной графической среды для компьютерного моделирования ТПС осуществляется в результате выполнения следующих трех этапов:

- 1) построение инструментальной подсистемы;
- 2) построение подсистемы доступа к базам данных;
- 3) построение графической подсистемы.

На первом и втором этапах выполняется настройка среды на выбранный пользователем тип ТПС, а на третьем этапе — настройка на работу с конкретными особенностями моделируемой ТПС.

Рассмотрим подробнее содержание этих этапов.

Этап 1. Построение инструментальной подсистемы. На основе описания типа ТПС, соответствующего выбору пользователя, создаются элементы графического интерфейса и необходимые структуры данных.

Построение на первом этапе выполняется в соответствии со следующим алгоритмом.

1. Формирование структур данных, описывающих состав элементов ТПС конкретного типа и их свойства.

2. Создание главного меню графической среды, позволяющего пользователю решать задачи, соответствующие типу моделируемой ТПС.

3. Загрузка в память графических изображений для создания элементов панелей инструментов на основе описания элементов ТПС в онтологии ТПС.

4. Создание панелей инструментов, содержащих элементы графического интерфейса для работы с компьютерной моделью ТПС.

5. Формирование структур данных, описывающих список возможного к установке в ТПС оборудования и набор моделей (графических и математических).

6. Формирование структур данных, описывающих состав решаемых прикладных задач и используемого для их решения ПО.

7. Формирование структур данных, описывающих список входных и выходных параметров для решаемых прикладных задач.

Структура формируемой инструментальной подсистемы представлена на рис. 5.

Этап 2. Построение подсистемы доступа к базам данных. Компьютерные модели ТПС различных типов хранятся в БД, состав объектов которых соответствует составу элементов ТПС, их свойствам, набору используемого оборудования и его характеристикам. Доступ к этим моделям осуществляется через ориентированные на конкретный тип ТПС специализированные программные компоненты. Подключение этих компонентов выполняется в автоматизированном режиме, во время которого из онтологии ПО считывается описание программных компонентов, соответствующих типу моделируемой ТПС. Далее по описанию этих компонентов выполняется их интеграция в программную систему средствами рефлексивного программирования.

Этап 3. Построение графической подсистемы. Основная идея, предлагаемая авторами, заключается в том, что графическая подсистема, от-



Рис. 5. Структура инструментальной подсистемы

вещающая за отображение активной модели ТПС, строится динамически (в автоматизированном режиме) в ответ на действия пользователя, направленные на внесение изменений в компьютерную модель ТПС. В случае, когда пользователь создает новую модель ТПС определенного типа, интегрированная графическая среда создает базовый набор моделей элементов, соответствующих типу моделируемой ТПС. В случае, когда пользователь загружает ранее созданную модель ТПС, среда дополнительно к базовому набору моделей элементов ТПС подключает набор моделей элементов, соответствующий составу элементов моделируемой ТПС. Модель элемента (компонент-модель) представляет собой программный компонент, реализующий графическое изображение, и набор алгоритмов, необходимых для обеспечения активности компьютерной модели ТПС (например, обработка действий пользователя мышью, проверка принадлежности точки области элемента, перемещение, масштабирование и др.).

Когда пользователь добавляет новый элемент к модели ТПС, то этот элемент отображается в базовом состоянии, например, для ветви это будет трубопровод в активном состоянии. Если пользователь изменит состояние элемента или укажет установленное на этой ветви оборудование, то среда автоматически подключит соответствующую модель элемента и отобразит ее в составе компьютерной модели ТПС. Например, если пользователь укажет, что на ветви установлен регулятор давления типа "до себя", то соответствующий компонент-модель из библиотеки компонентов будет интегрирован в программную систему.

Каждый элемент моделируемой ТПС представляется отдельным программным компонентом и обладает уникальным номером. Все компоненты, реализующие узлы (источники, узлы-соединения, потребители), являются экземплярами класса *Node*, все компоненты, реализующие ветви (трубопроводы, насосные станции, ветви с регуляторами давления и расхода) — экземплярами класса *Branch*. Хранение в памяти этих компонентов организовано с помощью хэш-таблиц. Схема процесса построения графической подсистемы представлена на рис. 6.

Когда пользователь создает новый элемент компьютерной модели ТПС, менеджер моделей

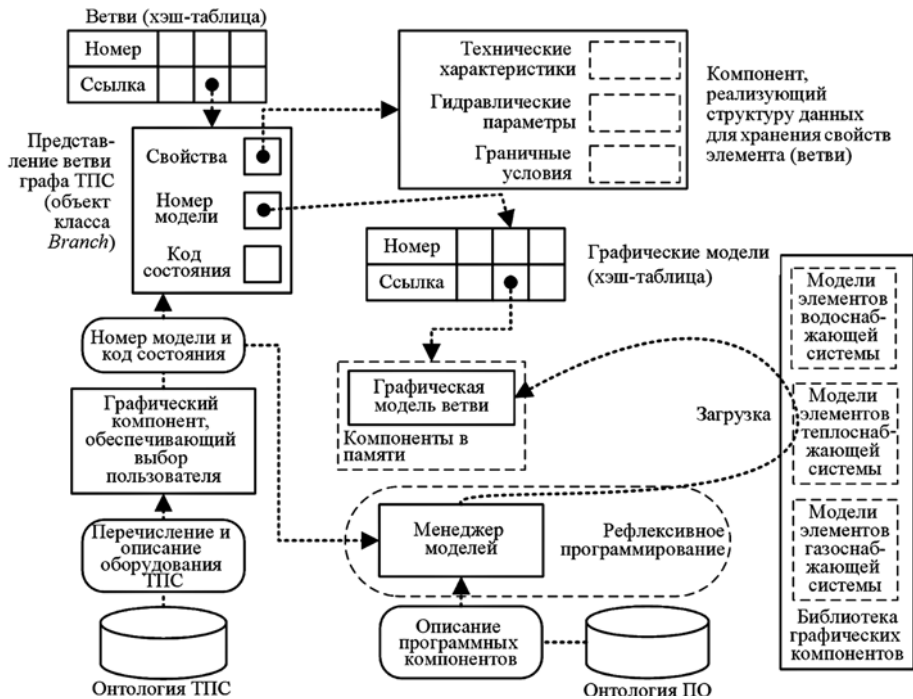


Рис. 6. Схема процесса построения графической подсистемы

выделяет память для хранения структуры данных, описывающей этот элемент. Далее пользователь заносит данные о свойствах этого элемента ТПС через соответствующие средства графического интерфейса. После этого пользователь выбирает модель, описывающую характеристики оборудования, и устанавливает состояние элемента. Менеджер моделей проверяет, подключен ли необходимый компонент-модель, реализующий выбранную пользователем модель элемента ТПС. Если необходимая модель не является доступной, то она интегрируется в программную систему из библиотеки графических компонентов средствами рефлексивного программирования. Когда модель становится доступной, то в структуру данных, описывающую новый элемент ТПС, заносится номер этой модели. Однажды подключенный к программной системе компонент-модель может быть многократно использован для представления элементов ТПС, тип и состояние которых соответствует этой модели.

Практическая реализация

Разработанный методический подход применен при реализации программной системы, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС. Реализованный программный прототип ориентирован на работу с компьютерными моделями теплоснабжающих

систем и позволяет создавать, вносить изменения и сохранять их в БД для многократного использования при решении прикладных задач.

Подсистемы интегрированной графической среды, отвечающие за создание инструментов графического интерфейса (инструментальной подсистемы), взаимодействие с БД (подсистемы доступа к базам данных) и работу с активной моделью ТПС (графической подсистемой), формируются в автоматизированном режиме с учетом особенностей конкретного типа моделируемой ТПС. Построение выполняется в соответствии с представленной в настоящей статье методикой.

На рис. 7 представлен вид главного окна интегрированной графической среды, которое содержит элементы интерфейса, формируемые инструментальной и графической подсистемами. Инструментальная подсистема обеспечивает работу следующих элементов главного окна интегрированной графической среды:

- главное меню, которое позволяет пользователю выполнить следующие действия: работать с файлом БД; управлять отображением модели ТПС; управлять видом рабочей области окна; решать прикладные задачи моделирования ТПС;
- панели инструментов, которые обеспечивают пользователю реализацию следующих функций: работа с файлом БД; управление отображением модели ТПС; управление видом рабочей области окна; работа с элементами активной графической модели ТПС; работа с объектами городской застройки на плане местности;
- дерево схемы моделируемой ТПС для работы с ее элементами;
- строка состояния, которая отображает основные сведения о процессе моделирования ТПС.

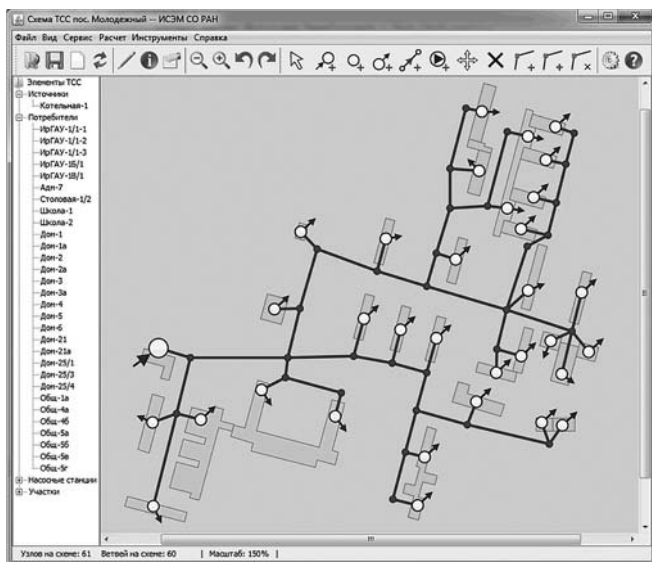


Рис. 7. Вид графического интерфейса интегрированной среды

Центральную часть главного окна занимает рабочая область, отвечающая за работу пользователя с активной графической моделью ТПС и объектами городской застройки на плане местности. За реализацию этой части пользовательского интерфейса отвечает графическая подсистема, которая динамически адаптируется (формируется) к следующим действиям пользователя:

- загрузка в память данных, описывающих ранее созданную модель ТПС;
- создание новой модели ТПС при выборе пользователем ее типа;
- изменения, вносимые пользователем в компьютерную модель ТПС.

Заключение

Предложен авторский подход к автоматизированному построению интегрированной графической среды для компьютерного моделирования ТПС различного типа. Оригинальность этого подхода заключается в реализации следующих идей:

- компьютерная модель ТПС определенного типа может быть представлена как совокупность графа, описывающего конфигурацию этой системы, и набора программных компонентов-моделей, описывающих свойства ее элементов;
- автоматизированное построение программной системы, реализующей интегрированную графическую среду, выполняется на основе компьютерной модели ТПС и онтологий на базе концепции Model-Driven Engineering и рефлексивного программирования.

Авторами реализована программная система, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС энергетики различного типа и назначения.

Предложенный авторами методический подход обладает универсальным характером и может быть использован в организациях, занимающихся разработкой ПО для компьютерного моделирования ТПС энергетики различного типа и назначения.

Список литературы

1. Меренков А. П., Хасилев В. Я. Теория гидравлических цепей. М.: Наука, 1985. 280 с.
2. Коголовский М. Р. Перспективные технологии информационных систем. М.: ДМК-Пресс, 2003. 288 с.
3. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering #1. USA, Morgan & Claypool, 2012. 182 p.
4. Völter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. USA, Wiley, 2006. 446 p.

5. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение концепции Model-Driven Engineering в программном комплексе для определения оптимальных параметров теплоснабжающих систем // Программная инженерия. 2016. Т. 7, № 3. С. 108—116.

6. Forman I., Forman N. Java Reflection in Action. Greenwich: Manning Publications Co., 2005. 273 p.

7. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение метапрограммирования в программном комплексе для решения задач схемно-параметрической оптимизации теплоснабжающих систем // Программная инженерия. 2011. № 6. С. 31—35.

8. Gosling J., Joy B., Steele G., Bracha G., Buckley A. Java Language Specification, Java SE 8 Edition. USA, Addison-Wesley Professional, 2014. 792 p.

V. A. Stennikov, Corresponding Member of the Russian Academy of Sciences,

Professor, Honored Science Worker of the Russian Federation, Director, e-mail: sva@isem.irk.ru,

E. A. Barakhtenko, Ph. D. in Technical Sciences, Senior Researcher, e-mail: barakhtenko@isem.irk.ru,

D. V. Sokolov, Ph. D. in Technical Sciences, Senior Researcher, e-mail: sokolov_dv@isem.irk.ru,

Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences (ESI SB RAS)

A Methodology for the Creation of an Integrated Graphical Environment for Computer Modeling of Energy Pipeline Systems

This paper presents a new methodology for the creation of an integrated graphical environment for computer modeling of energy pipeline systems of various types. This environment is a software system for solving information, computational and analytical problems within a single user interface.

The developed methodology includes:

- principles of automated creation of an integrated graphical environment for computer modeling of energy pipeline systems;
- architecture of the integrated graphical environment, describing the design of the software system created in an automated mode;
- procedure for automated software system creation based on the Model-Driven Engineering paradigm, reflective programming, ontologies and computer model of the pipeline system.

The main ideas underlying the methodology are:

- The software system focused on modeling a particular pipeline system is created in an automated mode in the context of solving the problem of building a computer model of this pipeline system.
- The automated construction of the software system and its adaptation to specific features of the modeled pipeline system is performed on the basis of reflective programming.
- The knowledge describing common properties of all pipeline system types, problems and software, are stored as ontologies, and specific features of the modeled pipeline system are described by its computer model.
- The automation of stages of the software system creation, and design of the user interface, involves the knowledge stored as ontologies.

The implemented software system is a software prototype of an integrated graphical environment for computer modeling of energy pipeline systems.

The proposed methodology is universal and can be used in organizations engaged in the development of software for computer modeling of energy pipeline systems of various types and purposes.

Keywords: integrated graphical environment, methodology, automation of programming, Model-Driven Engineering, metaprogramming, reflective programming, ontology, pipeline system, power engineering, software engineering

References

1. Merenkov A. P., Khasilev V. Ya. *Teoriya gidravlicheskikh tsepei* (The Theory of Hydraulic Circuits), Moscow, Nauka, 1985, 280 p. (in Russian).

2. Kogalovskii M. R. *Perspektivnye tekhnologii informatsionnykh system* (Perspective Technologies of Information Systems), Moscow, DMK-Press, 2003, 288 p. (in Russian).

3. Brambilla M., Cabot J., Wimmer M. *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering No. 1*. San Rafael, Morgan & Claypool, 2012, 182 p.

4. Völter M., Stahl T., Bettin J., Haase A., Helsen S. *Model-Driven Software Development: Technology, Engineering, Management*. New York, Wiley, 2006, 446 p.

5. Stennikov V. A., Barakhtenko E. A., Sokolov D. V. *Primenenie kontseptsii Model-Driven Engineering v programmnom*

komplekse dlya opredeleniya optimal'nykh parametrov teplosnabzhayushchikh sistem (Model-Driven Engineering in the Software for Parameter Optimization of Heat Supply Systems), *Programmnyaya Inzheneriya*, 2016, vol. 7, no. 3, pp. 108—116 (in Russian).

6. Forman I., Forman N. *Java Reflection in Action*. Greenwich, Manning Publications Co., 2005, 273 p.

7. Stennikov V. A., Barakhtenko E. A., Sokolov D. V. *Primenenie metaprogramirovaniya v programmnom komplekse dlya resheniya zadach shemno-parametricheskoy optimizatsii teplosnabzhayushchikh sistem* (Metaprogramming in the software for solving the problems of heat supply system schematic and parametric optimization), *Programmnyaya Inzheneriya*, 2011, no. 6, pp. 31—35 (in Russian).

8. Gosling J., Joy B., Steele G., Bracha G., Buckley A. *Java Language Specification, Java SE 8 Edition*. Boston, Addison-Wesley Professional, 2014, 792 p.

А. С. Акопов^{1, 2}, д-р техн. наук, доц.,
проф. кафедры бизнес-аналитики¹, вед. науч. сотр.², e-mail: aakopov@hse.ru,
А. Л. Бекларян¹, канд. техн. наук, доц. кафедры бизнес-аналитики, e-mail: abeklaryan@hse.ru,
Н. К. Хачатрян^{1, 2}, канд. физ.-мат. наук,
доц. кафедры бизнес-аналитики¹, ст. науч. сотр.², e-mail: nerses@cemi.rssi.ru; nkhatatryan@hse.ru,
А. В. Фомин¹, канд. экон. наук, ст. преподаватель, e-mail: afomin@hse.ru,
¹Национальный исследовательский университет "Высшая школа экономики",
²Федеральное государственное бюджетное учреждение науки
Центральный экономико-математический Институт РАН

Разработка адаптивного генетического оптимизационного алгоритма с использованием методов агентного моделирования

Представлен новый подход к разработке адаптивного генетического оптимизационного алгоритма (MAGAMO/A) с использованием методов агентного моделирования. Особенностью данного подхода является поддержка механизма адаптивного управления ключевыми характеристиками ГА, в частности, значениями вероятностей операторов кроссинговера и мутации, их типами и другими важными характеристиками, влияющими на разнообразие популяции и скорость сходимости ГА. Поддержка адаптивного управления обеспечивается за счет использования механизма карт состояний агента и заданных правил перехода между соответствующими состояниями, определяющими значения управляющих параметров ГА на индивидуальном уровне каждого агента-процесса. Проведен обзор наиболее популярных ГА, применяемых для задач многокритериальной оптимизации, в том числе, таких как SPEA2, NSGA, MOEA и др. Рассмотрены основные метрики оценки эффективности подобных ГА (Hypervolume, Generational Distance, расстояния между решениями на границе Парето и др.). Продемонстрирована эффективность разработанного подхода в решении оптимизационных задач большой размерности на нескольких тестовых примерах и в сравнении с другими известными ГА. Сформулированы основные направления дальнейших исследований в области разработки агентно-ориентированных генетических алгоритмов.

Ключевые слова: генетический алгоритм, фронт Парето, эволюционные вычисления, агентное моделирование, многоцелевая оптимизация

Введение

Разработка эффективных алгоритмов поиска решений для многокритериальных оптимизационных задач большой размерности является весьма актуальным направлением. Высокая степень сложности таких задач во многом обусловлена значительными затратами, требуемыми для вычисления значений целевых функций и других модельных характеристик, участвующих в проверке соответствующих ограничений. В случае, когда целевой функционал может быть описан аналитически, для определения локальных экстремумов применяют различные градиентные методы поиска решений ненулевого порядка, например методы Ньютона и Коши и их различные

модификации. В противном случае применяют хорошо известные методы прямого поиска (т. е. без вычисления производных), такие как метод Хука—Дживса, метод Нелдера—Мида и др. При этом постановка задачи многоцелевой оптимизации предполагает определение набора наилучших (компромиссных) решений, при выборе которых относительный уровень снижения качества по одному или нескольким критериям не превосходит относительного уровня повышения качества по другим критериям. Таким образом, требуется нахождение решений, при которых значение каждого частного показателя, характеризующего систему, не может быть улучшено без ухудшения других показателей. Подобное множество решений относится к типу Парето-оптимальных. По-

строение приближенного множества (или некоторого репрезентативного подмножества) Парето относится к числу задач высокой вычислительной сложности [21].

Одним из наиболее эффективных методов решения подобных задач являются генетические оптимизационные алгоритмы (ГА), впервые предложенные в работах Дж. Холланда и Д. Гольберга [10, 13]. В настоящее время для решения задач многокритериальной оптимизации и построения фронта Парето применяют широкое семейство ГА, в том числе такие как NSGA-II, NSGA-III, OMOPSO, PAES, PESA-II, SPEA2, VEGA, MOEA/D, MAGAMO и др. [3, 4, 6, 8, 9, 19, 20]. Среди них следует выделить алгоритм SPEA2 (Strength Pareto Evolutionary Algorithm) [20], отличающийся учетом ранга Парето для отбираемых в процессе эволюционного поиска решений, а также относительно новый алгоритм MOEA/D [17], основанный на декомпозиции решаемой многоцелевой оптимизационной задачи на множество одноцелевых задач.

Важным преимуществом ГА по сравнению с другими методами многомерной оптимизации является возможность эффективного распараллеливания вычислительной процедуры [23–25]. При этом следует отметить два подхода. Первый, наиболее известный, подход основан на так называемой "островной" модели [17], когда из одной общей популяции возможных решений (удовлетворяющих правилам недоминирования Парето), именуемой "материком", выделяется несколько популяций меньшей длины, эволюция которых осуществляется параллельно с периодическим обменом наилучшими решениями с "материком". Достоинством данного подхода является существенное ускорение вычислительного процесса в условиях мультипроцессорной архитектуры. Однако с ростом размерности оптимизационной задачи, длина популяции решений начинает расти пропорционально числу переменных решений. Для преодоления данной проблемы нами ранее был предложен другой подход — алгоритм MAGAMO [4, 26], основанный на равномерном разбиении пространства решений между автономными агентами (являющимися самостоятельными ГА) в целях минимизации длины популяции на индивидуальном уровне агентов. Данный подход показал высокую эффективность для многокритериальных оптимизационных задач большой размерности, в которых целевые функционалы вычисляются в результате имитационного моделирования. Тем не менее вычислительная продуктивность MAGAMO была ограничена (например, вероятностью оператора мутации) и зависела в основном от значений параметров ГА, устанавливаемых равнозначными

для всех агентов-процессов. Для преодоления данной проблемы предлагается создание адаптивной модификации алгоритма MAGAMO/A с использованием возможностей методов агентного моделирования, в частности механизма карты состояний, являющейся, по сути, конечным автоматом для управления параметрами ГА на индивидуальном уровне в зависимости от качества эволюционного процесса каждого взаимодействующего агента.

Основной целью данной работы является разработка многоагентного адаптивного генетического оптимизационного алгоритма, обеспечивающего эффективный процесс коэволюционного поиска в условиях мультипроцессорной системной архитектуры для многокритериальных оптимизационных задач большой размерности (несколько целевых функций, сотни переменных, множественные ограничения). Данный подход был успешно апробирован для решения сложных эколого-экономических оптимизационных задач [1, 2], для управления поведением агентов-спасателей при чрезвычайных ситуациях [5] и др.

1. Концептуальная модель многоагентного ГА

Примером использования методов агентного моделирования для построения генетического оптимизационного алгоритма является работа [4]. В данной работе был предложен многоагентный генетический алгоритм MAGAMO, основанный на принципе распределения пространства решений между агентами-процессами и организацией коэволюционной процедуры, в рамках которой центральный процесс отвечает за формирование Парето-оптимальных решений, а локальные процессы обеспечивают направленный поиск наилучших решений для назначенных их переменных. Эффективность подобного алгоритма достигалась за счет существенного снижения длины популяции (и, как следствие, сокращения временных затрат на оценку фитнес-функций) на уровне локальных агентов-процессов и обеспечения непрерывного обмена значениями наилучших недоминируемых решений с глобальным архивом.

Отметим, что принципы построения MAGAMO существенно отличаются от классической "островной" модели, в которой нет распределения искомым переменных между процессами, а эффективность достигается по сути за счет более диверсифицированной эволюции. Кроме того, в MAGAMO есть существенные отличия от "роевых" алгоритмов (*swarm optimization* [16]), функциональность которых основана на движении частиц в пространстве решений и

обмене полезной информацией между ними. В частности, число агентов в MAGAMO, как правило, выбирается равным числу процессоров, на которых могут быть запущены исполняемые потоки. В "роевых" алгоритмах число агентов-частиц может измеряться тысячами, что и позволяет достичь наибольшей эффективности за счет максимального быстрого сканирования ландшафта целевых функционалов и определения целевой точки. Однако временные затраты на обмен информацией между агентами-частицами в большой популяции и отсутствие операторов мутации существенно ограничивают применимость подобных алгоритмов к сложным задачам многоцелевой оптимизации, требующим определения фронта Парето (в особенности, имеющего множественные разрывы, являющегося невыпуклым и др.). Поэтому предлагается, сохранив базовую функциональность MAGAMO, обеспечить его максимальную эффективность за счет адаптивного управления параметрами на индивидуальном уровне агентов-процессов. Для этого предлагается изменять значение одного или нескольких параметров ГА в зависимости от качества эволюционного процесса — темпов роста приспособленности популяции и степени разнообразия генотипов популяции (удаленности решений друг от друга участвующих в селекции). Данный подход экспериментально демонстрирует свою эффективность. Однако, когда популяция решений эволюционирует в рамках одного процесса, определить наилучшие значения параметров ГА, являющиеся рациональными для всех решений, не всегда удастся ввиду их неоднородности (значения одних переменных требуют более быстрой эволюции, другие должны лишь немного корректироваться и т. д.). Потому более эффективным представляется подход, основанный на принципах адаптивного управления параметрами ГА на индивидуальном уровне агентов-процессов. Отметим, что преимущества адаптивных ГА ранее были исследованы в работе [22].

Для разработки адаптивных ГА можно использовать систему управления состояниями каждого агента-процесса на основе метода конечных автоматов (FSM) реализуемого посредством карт состояний агента (*statecharts*). Переходы между состояниями каждого агента-процесса осуществляются в зависимости от текущей эффективности ГА, оцениваемой с использованием различных критериев, например, критерия "гиперобъема" (*Hypervolume*), расстояния между решениями на фронте Парето GD, IGD, Spacing и другие метрики [7, 11, 14].

Отметим, что данный подход является эффективным при значительном времени эволюционного процесса, когда возможны множествен-

ные изменения состояний и параметров ГА, т. е. именно для оптимизационных задач большой размерности. На начальном шаге все искомые переменные равномерно распределяются между агентами-процессами и далее обмениваются друг с другом информацией о наилучших (с точки зрения приспособленности фитнес-функции) значениях собственных переменных, удовлетворяющих критерию Парето-оптимальности.

2. Формальное описание адаптивного генетического алгоритма

Итак, рассматривается ансамбль взаимодействующих агентов-процессов, являющихся автономными ГА и имеющих следующие возможные состояния для каждого агента.

1. **Состояние 0.** *Начальная локальная эволюция* — состояние агента-процесса для первых стадий ГА, при котором применяются стандартные значения параметров ГА. В данном состоянии обмен наилучшими решениями (значениями искомым переменных) с другими агентами-процессами осуществляется на каждой итерации (эпохе) ГА.

2. **Состояние 1.** *Быстрая эволюция* — состояние агента-процесса для последующих эпох ГА, при котором применяются параметры ГА, обеспечивающие максимальные темпы эволюционного процесса, например, двухточечный оператор кроссинговера; три-четыре родительских особи; множество генерируемых потомков; почти нулевая вероятность оператора мутации и др. В данном состоянии обмен наилучшими решениями с другими агентами-процессами осуществляется на каждой итерации ГА. Отметим, что рассматриваемое состояние может привести к преждевременной сходимости ГА. Кроме того, данное состояние требует постоянных коммуникационных затрат на взаимодействие с другими агентами, что влияет на скорость эволюционных вычислений.

3. **Состояние 2.** *Медленная эволюция с низким разнообразием популяции*, — состояние агента-процесса для последующих эпох ГА, при котором применяются параметры ГА, обеспечивающие низкие темпы эволюционного процесса, например, одноточечный оператор кроссинговера, две родительских особи с двумя потомками, небольшая вероятность оператора мутации и др. При входе и выходе из данного состояния осуществляется обмен наилучшими решениями с другими агентами. Отметим, что данное состояние может привести к замедлению работы ГА и предотвращению преждевременной сходимости за счет оператора мутации.

4. **Состояние 3.** *Медленная эволюция с высоким разнообразием популяции* — состояние агента-

Значения основных параметров ГА в зависимости от состояния агента-процесса

Параметры ГА	Состояния ГА			
	0	1	2	3
Вероятность оператора кроссинговера	1	0,85	0,5	0,5
Вероятность оператора мутации	0	0,001	0,01	0,2
Тип оператора кроссинговера	Одноточечный	Двухточечный	Одноточечный	Равномерный
Тип оператора мутации	Нет	Стандартный	Стандартный	Комбинированный
Число родительских особей	2	3 и более	2	3 и более
Число особей-потомков	2	3 и более	2	3 и более
Максимальное число эпох ГА, допустимых для данного состояния	100	1000	300	100
Частота обмена наилучшими решениями с другими агентами-процессами	На каждой итерации	На каждой итерации	Только на входе и выходе из состояния	Только на входе и выходе из состояния

процесса для последующих эпох ГА, при котором применяются параметры ГА, обеспечивающие низкие темпы эволюционного процесса, но наибольший уровень диверсификации популяции за счет множественных (комбинированных) мутаций. При входе и выходе из данного состояния осуществляется обмен наилучшими решениями с другими агентами. Отметим, что данное состояние гарантированно обновляет популяцию, но также может привести к редукции ГА (т. е. к разрушению приспособленности популяции) в случае длительного нахождения ГА в данном режиме.

5. Состояние 4. Завершение локальной эволюции — состояние агента-процесса, при котором найдено подмножество наилучших значений искомым переменных, принадлежащих данному агенту и удовлетворяющих критерию оптимальности по Парето (т. е. решений, имеющих наивысший ранг Парето). Одним из условий перехода к данному состоянию является стабилизация средней приспособленности локальной популяции (т. е. она становится однородной) до требуемого уровня сходимости. В случае невозможности достижения глобальной сходимости в течение значительного числа эпох ГА (например, более 1000), возможен возврат к состоянию быстрой эволюции.

6. Состояние 5. Окончание глобальной эволюции — состояние агента-процесса, при котором найдены окончательные наилучшие значения искомым переменных, принадлежащих всем агентам-процессам.

Значения параметров ГА, соответствующие первым четырем состо-

стояниям, представлены в табл. 1. Последние два состояния характеризуются отсутствием эволюционного процесса.

На рис. 1 представлена карта состояний агента-процесса, реализованная в системе AnyLogic, используемой для реализации адаптивного многоагентного генетического алгоритма (MAGAMO/A) на языке программирования Java.

Приведем формальное описание правил перехода к соответствующим состояниям для предложенного генетического алгоритма.

Введем следующие обозначения:

- $t \in \{t_0, t_0 + 1, \dots, t_0 + T\}$ — индекс эпох (внутреннее время) ГА, T — максимальное допустимое число эпох ($T = 10\,000$);

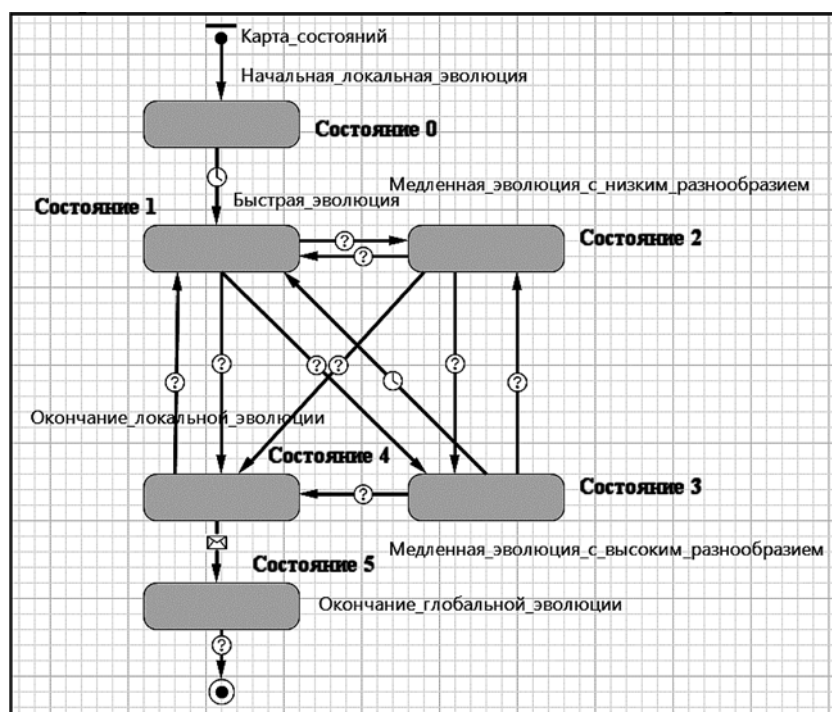


Рис. 1. Карта состояний агента-процесса (генетического алгоритма MAGAMO/A)

- $i \in \{1, 2, \dots, I\}$ — индекс агента-процесса;
- $st_i(t) \in \{0, 2, \dots, 5\}$ — возможные состояния агента-процесса (рис. 1).
- $HV_i(t)$ — среднее (медианное) значение метрики "гиперобъема", вычисляемое на основе оценки площади над аппроксимируемым фронтом Парето, ограниченным точкой надира с известными координатами (т. е. соответствующей наихудшим значениям целевых функционалов оптимизационной задачи, рис. 2). Отметим, что данный индикатор является наиболее важным при оценке эффективности ГА для задач многокритериальной оптимизации.
- HV — минимально допустимое значение метрики "гиперобъема";
- $IGD_i(t)$ — среднее (медианное) значение инвертированного расстояния от каждого Парето-оптимального решения к ближайшему решению аппроксимируемого фронта Парето (рис. 3);
- IGD — максимально допустимое значение инвертированного расстояния между решениями.

Отметим, что для стандартных задач минимизации целевых функционалов (см. рис. 2), значения метрики "гиперобъема" максимизируют (т. е. чем дальше мы от наихудшего сценария, тем лучше).

Инвертированное расстояние между решениями (рис. 3), лежащими на аппроксимируемом фронте Парето, и ближайшими решениями из сформированного множества решений является одной из метрик качества решений, что обусловлено важным требованием к аппроксимируемому фронту Парето — необходимости наличия близко расположенных (к фактическому подмножеству) решений.

- $SP_i(t)$ — среднее (медианное) значение расстояния между Парето-оптимальными решениями, лежащими на фронте (рис. 4);
- SP — минимально допустимое значение расстояния между решениями, лежащими на фронте Парето.

Минимально допустимый темп роста приспособленности популяции (минимально допустимая скорость эволюции) равна

$$F_i(t) = \frac{1}{t} \sum_{\zeta=t_0}^t \tilde{f}_i(\zeta), \quad (1)$$

$$t \in \{t_0, t_0 + 1, \dots, t_0 + T\}, i \in \{1, 2, \dots, I\},$$

где ζ — индекс текущей итерации ГА, $\zeta \in \{t_0, t_0 + 1, \dots, t_0 + T\}$; $\tilde{f}_i(\zeta)$ — среднее значение приспособленности популяции в момент времени ζ , вычисляемое на основе оценки значений фитнес-функций для всех решений — членов данной популяции;

- $G_i(t)$ — сходимость (конвергенция) i -го агента-процесса (локального ГА), оцениваемая как среднее (медианное) расстояние Хэмминга между членами популяции;
- \bar{G} — требуемый уровень локальной сходимости (малое число), при котором останавливается работа i -го агента-процесса (локального ГА), т. е. $G_i(t) \leq \bar{G}$ при условии, что $t > t^*$, где t^* — минимально допустимое время (число эпох ГА), необходимое для предотвращения преждевременной сходимости ($t^* = 100$);

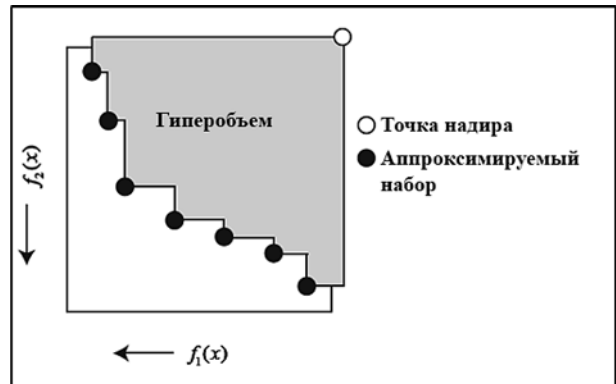


Рис. 2. Оценка метрики Hypervolume на примере задачи минимизации двух целевых функционалов

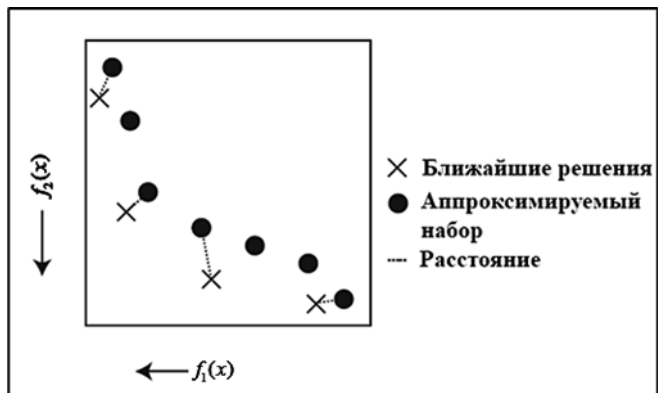


Рис. 3. Инвертированное расстояние между решениями

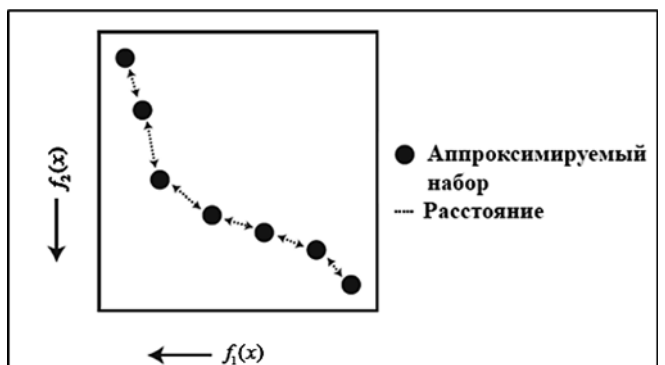


Рис. 4. Расстояние между решениями, лежащими на фронте Парето

- \bar{G} — требуемый уровень глобальной сходимости (малое число) при котором останавливается работа всех агентов-процессов;
- $\tau_{i1}(t)$, $\tau_{i2}(t)$, $\tau_{i3}(t)$ — максимально допустимое время нахождения агента-процесса в первом, втором и третьем состоянии соответственно.

Тогда правило перехода к новым состояниям i -го агента-процесса ($i \in \{1, 2, \dots, I\}$) в рамках алгоритма MAGAMO/A в момент времени $t \in \{t_0, t_0 + 1, \dots, t_0 + T\}$ может быть представлено в следующей форме:

$$st_i(t) = \begin{cases} 0, & \text{если выполняется I,} \\ 1, & \text{если выполняется II,} \\ 2, & \text{если выполняется III,} \\ 3, & \text{если выполняется IV,} \\ 4, & \text{если выполняется V,} \\ 5, & \text{если выполняется VI,} \end{cases} \quad (2)$$

где

- I. $t \leq t^*$ или $st_i(t) \notin \{1, 2, 3, 4, 5\}$;
- II. $t^* < t \leq \tau_{i1}(t)$ и $F \leq F_i(t) \leq \bar{F}$ и $HV_i(t) \geq \underline{HV}$ и $IGD_i(t) \leq \overline{IGD}$ и $SP_i(t) \geq \underline{SP}$ и $G_i(t) > \bar{G}$ и $st_i(t) \notin \{0, 5\}$;
- III. $t \leq \tau_{i2}(t)$ и $F_i(t) > \bar{F}$ и $HV_i(t) \geq \underline{HV}$ и $IGD_i(t) \leq \overline{IGD}$ и $SP_i(t) \geq \underline{SP}$ и $G_i(t) > \bar{G}$ и $st_i(t) \notin \{0, 4, 5\}$;
- IV. $t \leq \tau_{i3}(t)$ и $(F_i(t) > \bar{F})$ или $HV_i(t) < \underline{HV}$ или $IGD_i(t) > \overline{IGD}$ или $SP_i(t) < \underline{SP}$ и $G_i(t) > \bar{G}$ и $st_i(t) \notin \{0, 4, 5\}$;
- V. $G_i(t) \leq \bar{G}$ и $\sum_{i=1}^I G_i(t) > \bar{G}$ и $st_i(t) \notin \{0, 5\}$;
- VI. $\sum_{i=1}^I G_i(t) \leq \bar{G}$.

Далее с использованием системы AnyLogic был реализован прототип адаптивного много-агентного генетического алгоритма с использованием диаграммы действий **actionChart** (рис. 5).

Важной особенностью предложенного алгоритма (рис. 5) является поддержка механизма перехода к новым состояниям агента по результатам оценки качества эволюционного процесса (метрики "гиперобъема", расстояния между решениями на фронте Парето и др.), что обеспечивает возможность адаптивного управления параметрами ГА и повышает эффективность генетического алгоритма.

3. Результаты оптимизационных экспериментов

Отметим, что формальное описание тестовых многокритериальных оптимизационных задач представлено в работе [18]. В табл. 2 полужирным шрифтом выделены результаты (медианное значение метрики "гиперобъем"), соответствующие наилучшим значениям, полученные посредством множественных (50) прогонов ГА. Также отметим, что время, выделяемое для оценки данной метрики для всех рассматриваемых ГА было одинаковым (для сопоставимости оценки качества фронта Парето). Тем не менее временная эффективность (время достижения сходимости) у рассматриваемых ГА различная (указана в скобках). Все расчеты были проведены на персональном компьютере Intel Core i7-2670QM CPU@ 2.20GHz, 8Gb RAM. Расчеты для алгоритмов ϵ -MOEA, NSGAII и SPEA2 выполнялись с использованием программного продукта MOEAFramework [13]. Расчеты для алгоритма MAGAMO/A выполнялись с использованием Java-классов, созданных в AnyLogic и импортированных в IntelliJ IDEA для организации параллельных вычислений.

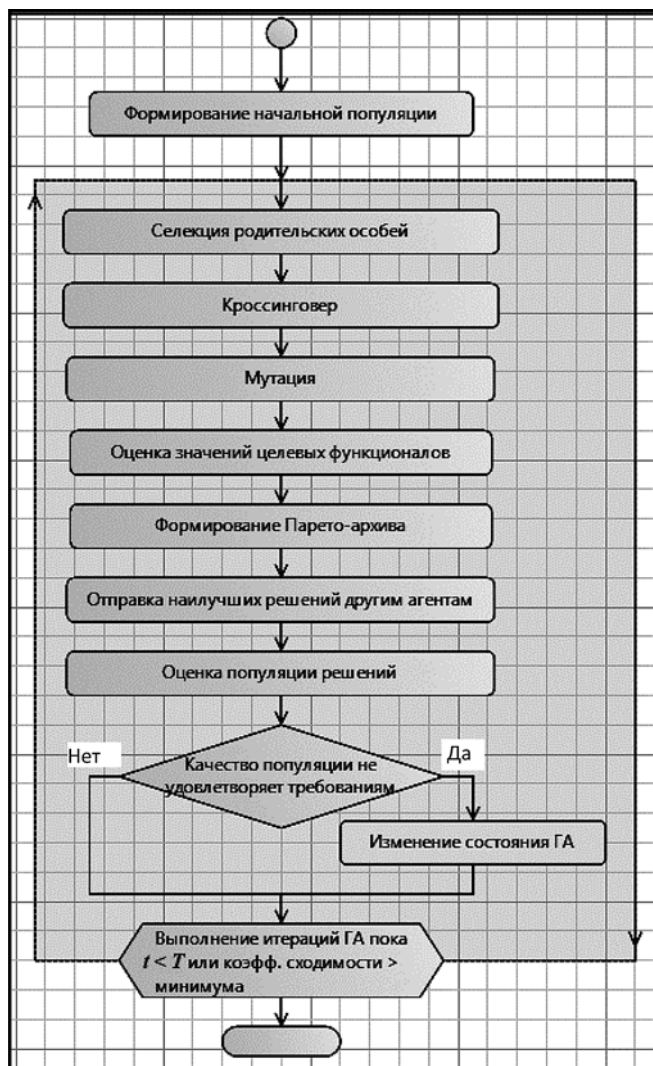


Рис. 5. Алгоритм агента-процесса (локальный ГА) в рамках модели MAGAMO/A

Оценка метрики "гиперобъем" (Hypervolume)

Тестовая оптимизационная задача	MAGAMO/A	ϵ -MOEA	NSGAII	SPEA2
UF1 (30 переменных, 2 целевых функционала)	Min: 0.364 Median: 0.461 Max: 0.535 (30 c)	Min: 0.321 Median: 0.459 Max: 0.539 (34 c)	Min: 0.136 Median: 0.479 Max: 0.544 (39 c)	Min: 0.416 Median: 0.515 Max: 0.545 (84 c)
DTLZ2_2 (11 переменных, 2 целевых функционала)	Min: 0.212 Median: 0.211 Max: 0.213 (20 c)	Min: 0.213 Median: 0.213 Max: 0.214 (23 c)	Min: 0.212 Median: 0.212 Max: 0.213 (29 c)	Min: 0.213 Median: 0.213 Max: 0.214 (232 c)
DTLZ7_5 (24 переменных, 5 целевых функционалов)	Min: 0.671 Median: 0.698 Max: 0.761 (22 c)	Min: 0.386 Median: 0.428 Max: 0.745 (26 c)	Min: 0.602 Median: 0.657 Max: 0.695 (44 c)	Min: 0.608 Median: 0.672 Max: 0.708 (426 c)
DTLZ7_7 (26 переменных, 7 целевых функционалов)	Min: 0.381 Median: 0.551 Max: 0.668 (24 c)	Min: 0.155 Median: 0.184 Max: 0.305 (26 c)	Min: 0.379 Median: 0.501 Max: 0.627 (43 c)	Min: 0.135 Median: 0.469 Max: 0.655 (731 c)
ZDT5 (80 переменных, 2 целевых функционала)	Min: 0.843 Median: 0.855 Max: 0.891 (11 c)	Min: 0.777 Median: 0.843 Max: 0.872 (17 c)	Min: 0.841 Median: 0.850 Max: 0.873 (22 c)	Min: 0.842 Median: 0.850 Max: 0.881 (390 c)

Из табл. 2 следует, что для многокритериальных оптимизационных задач большой размерности среди сравниваемых алгоритмов наиболее эффективным (как по критерию временной эффективности, так и по критерию "гиперобъема") является предложенный адаптивный многоагентный генетический алгоритм MAGAMO/A.

Следует отметить, что тестовые оптимизационные задачи DTLZ7_5, DTLZ7_7 и ZDT5 являются достаточно сложными, например, для задачи DTLZ7_7, характеризующейся наличием 26 искоемых переменных и 7 целевых функционалов, формируемый Парето-фронт имеет разрывы.

Отметим, что при решении стандартных малоразмерных многокритериальных оптимизационных задач целесообразность использования MAGAMO/A недостаточно обоснована. При этом наиболее близко по эффективности к нему находится алгоритм ϵ -MOEA, предложенный в работе [10]. Однако недостатком последнего является необходимость подбора индивидуальных значений параметра ϵ для каждой решаемой оптимизационной задачи. Для малоразмерных многокритериальных оптимизационных задач целесообразно использовать алгоритм NSGA-II [9] и его современные модификации NSGA-III [8], RVEA [6] и др.

Заключение

В данной работе предложен новый подход к разработке адаптивного генетического оптимизационного алгоритма с использованием методов агентного моделирования. Особенностью данного подхода является поддержка механизма адап-

тивного управления ключевыми характеристиками ГА, в частности, значениями вероятностей операторов кроссинговера и мутации, их типами и другими важными характеристиками ГА, влияющими на разнообразие популяции, скорость сходимости ГА, качество Парето-оптимальных решений и др. В результате проведенных численных экспериментов продемонстрирована высокая эффективность предложенного алгоритма (MAGAMO/A) для задач многокритериальной оптимизации большой размерности.

Дальнейшим направлением исследования является реализация предложенного алгоритма с использованием MPI на языке программирования C++, интеграция MAGAMO/A с имитационными моделями агентного типа, применение технологии CUDA [24] для организации эффективных взаимодействий между агентами-процессами при параллельных вычислениях и др.

Исследование выполнено за счет гранта Российского научного фонда (проект № 17-71-10116).

Список литературы

1. Akopov A. S., Beklaryan L. A., Saghatelyan A. K. Agent-based modelling for ecological economics: A case study of the Republic of Armenia // Ecological Modelling. 2017. Vol. 346. P. 99–118.
2. Akopov A. S., Beklaryan A. L., Saghatelyan A. K., Sahakyan L. V. Control system for ecological modernization of enterprises (on the example of the Republic of Armenia) // Business Informatics. 2016. N. 2 (36). P. 71–78.
3. Akopov A. S. Parallel genetic algorithm with fading selection // International Journal of Computer Applications in Technology. 2014. Vol. 49. N. 3/4. P. 325–331.
4. Akopov A. S., Hevencev M. A. A Multi-agent genetic algorithm for multi-objective optimization // Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2013. Manchester: IEEE, 2013. P. 1391–1395.

5. **Beklaryan A. L., Akopov A. S.** Simulation of Agent-rescuer Behaviour in Emergency Based on Modified Fuzzy Clustering // AAMAS'16: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. Richland: International Foundation for Autonomous Agents and Multiagent Systems, 2016. P. 1275–1276.
6. **Cheng R., Jin Y., Olhofer M., Sendhoff B.** A reference vector guided evolutionary algorithm for many-objective optimization // IEEE Transactions on Evolutionary Computation. 2016. Vol. 20. N. 5. P. 773–791.
7. **Coello Coello C. A., Lamont G. B., Van Veldhuizen D. A.** Evolutionary Algorithms for Solving Multi-Objective Problems. New York: Springer Science + Business Media LLC, NY, 2007. 800 p.
8. **Deb K., Jain H.** An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints // IEEE Transactions on Evolutionary Computation. 2014. Vol. 18, N. 4. P. 577–601.
9. **Deb K., Pratap A., Agarwal S., Meyarivan T.** A fast elitist multi-objective genetic algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation. 2000. Vol. 6, N. 2. P. 182–197.
10. **Deb K., Mohan M., Mishra S.** A fast multi-objective evolutionary algorithm for finding well-spread Pareto-optimal solutions. KanGAL Report No. 2003002, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, 2003.
11. **Knowles J., Corne D.** On metrics for comparing non-dominated sets // Proc. of Congress on Evolutionary Computation (CEC 2002), Honolulu, HI, 2002. P. 711–716.
12. **Goldberg D. E.** Genetic Algorithms in Search, Optimization and Machine Learning (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
13. **Hadka D.** Beginner's Guide to the MOEA Framework. Great Space Independent Publishing Platform, 2017. 214 p.
14. **Hadka D., Reed P.** Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization // Evolutionary Computation, 2012. Vol. 20. N. 3. P. 423–452.
15. **Holland J. H.** Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975.
16. **Poli R.** Analysis of the publications on the applications of particle swarm optimisation // Journal of Artificial Evolution and Applications. 2008. Vol. 2008. P. 1–10.
17. **Whitley D., Rana S., Heckendorn R. B.** The Island Model Genetic Algorithm: On Separability, Population Size and Convergence // Journal of Computing and Information Technology. 1998. Vol. 7. P. 33–47.
18. **Zhang Q., Zhou Q., Zhao S., Suganthan P. N., Liu W., Tiwari S.** Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, 2009.
19. **Zhang Q., Liu W., Li H.** The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances // Proc. of Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway, 2009. P. 203–208.
20. **Zitzler E., Laumanns M., Thiele L.** SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization. Barcelona, Spain, International Center for Numerical Methods in Engineering (CIMNE), 2002.
21. **Березкин В. Е., Каменев Г. К., Лотов А. В.** Гибридные адаптивные методы аппроксимации невыпуклой многомерной границы Парето // Журнал вычислительной математики и математической физики. 2006. Т. 46, № 11. С. 2009–2023.
22. **Зинченко Л. А., Курейчик В. М.** Многоальтернативные генетические алгоритмы поиска экстремума функции // Известия Российской академии наук. Теория и системы управления. 2003. № 4. С. 82–91.
23. **Кныш Д. С., Курейчик В. М.** Параллельные генетические алгоритмы: обзор и состояние проблемы // Известия Российской академии наук. Теория и системы управления. 2010. № 4. С. 72–82.
24. **Лунин Д. В., Скворцов С. В.** Разработка параллельного генетического алгоритма для решения задачи коммивояжера на платформе CUDA // Системы управления и информационные технологии. 2015. Т. 60. № 2. С. 50–55.
25. **Мокшин В. В., Якимов И. М.** Метод формирования модели анализа сложной системы // Информационные технологии. 2011. № 5. С. 46–51.
26. **Хивинцев М. А., Акопов А. С.** Применение многоагентного генетического алгоритма для поиска оптимальных стратегических и оперативных решений // Бизнес-информатика. 2014. № 1 (27). С. 23–33.

A. S. Akopov, D. Sc. Professor, e-mail: aakopov@hse.ru,
 National Research University Higher School of Economics, Moscow,
 Central Economics and Mathematics Institute RAS, Moscow,
A. L. Beklaryan, Ph. D., Associate Professor, e-mail: abeklaryan@hse.ru,
 National Research University Higher School of Economics, Moscow,
N. K. Khachatryan, Ph. D., Associate Professor, e-mail: nerses@cemi.rssi.ru, nkhachatryan@hse.ru,
 National Research University Higher School of Economics, Moscow, 101000,
 Central Economics and Mathematics Institute RAS, Moscow, 117418,
A. V. Fomin, Ph. D., Senior Lecturer, e-mail: afomin@hse.ru,
 National Research University Higher School of Economics, Moscow, 101000, Russian Federation

Development of an Adaptive Genetic Optimization Algorithm using Agent Modeling Methods

This article presents a new approach to developing an adaptive genetic optimization algorithm (MAGAMO/A) using agent modeling techniques. The peculiarity of this approach is the support of the mechanism of adaptive control of key characteristics of GA, in particular, the values of the probabilities of crossover operators and mutations, their types and other important characteristics that affect the population diversity and the rate of convergence of GA. Support for adaptive control is provided by using the mechanism of agent state charts and the specified rules of transition between the corresponding states that determine the values of the control parameters of the GA at the individual level of each agent-process. The review of the most popular GAs used for multicriteria optimization, including SPEA2, NSGA, MOEA, etc., is reviewed. The main metrics for evaluating the effectiveness of such GAs (Hypervolume, Generational Distance, distance between solutions on the Pareto boundary, etc.) are considered. The efficiency of the developed approach in the solution of optimization problems of large dimension on several test examples and in comparison with other known GA is demonstrated. The main directions of further research in the field of development of agent-oriented genetic algorithms are formulated.

Keywords: genetic algorithm, Pareto front, evolutionary computation, accentual modeling, multi-objective optimization

References

1. Akopov A. S., Beklaryan L. A., Saghatelian A. K. Agent-based modelling for ecological economics: A case study of the Republic of Armenia, *Ecological Modelling*, 2017, vol. 346, pp. 99–118.
2. Akopov A. S., Beklaryan A. L., Saghatelian A. K., Sahakyan L. V. Control system for ecological modernization of enterprises (on the example of the Republic of Armenia), *Business Informatics*, 2016, no. 2 (36), pp. 71–78.
3. Akopov A. S. Parallel genetic algorithm with fading selection, *International Journal of Computer Applications in Technology*, 2014, vol. 49, no. 3/4, pp. 325–331.
4. Akopov A. S., Hevencev M. A. A Multi-agent genetic algorithm for multi-objective optimization, in: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2013. Manchester, IEEE, 2013, pp. 1391–1395.
5. Beklaryan A. L., Akopov A. S. Simulation of Agent-rescuer Behaviour in Emergency Based on Modified Fuzzy Clustering, in: *AAMAS'16: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. Richland: International Foundation for Autonomous Agents and Multiagent Systems*, 2016, pp. 1275–1276.
6. Cheng R., Jin Y., Olhofer M., Sendhoff B. A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation*, 2016, vol. 20, no. 5, pp. 773–791.
7. Coello Coello C. A., Lamont G. B., Van Veldhuizen D. A. *Evolutionary Algorithms for Solving Multi-Objective Problems*, New York, Springer Science + Business Media LLC, NY, 2007, 800 p.
8. Deb K., Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, 2014, vol. 18, no. 4, pp. 577–601.
9. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast elitist multi-objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 2000, vol. 6, no. 2, pp. 182–197.
10. Deb K., Mohan M., Mishra S. *A fast multi-objective evolutionary algorithm for finding well-spread Pareto-optimal solutions*, KanGAL Report No. 2003002, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, 2003.
11. Knowles J., Corne D. On metrics for comparing non-dominated sets, in *Congress on Evolutionary Computation (CEC 2002)*, Honolulu, HI, 2002, pp. 711–716.
12. Goldberg D. E. *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
13. Hadka D. *Beginner's Guide to the MOEA Framework*. GreatSpace Independent Publishing Platform, 2017. 214 p.
14. Hadka D., Reed P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization, *Evolutionary Computation*, 2012, vol. 20, no. 3, pp. 423–452.
15. Holland J. H. *Adaptation in natural and artificial systems*, Ann Arbor, University of Michigan Press, 1975.
16. Poli R. Analysis of the publications on the applications of particle swarm optimisation, *Journal of Artificial Evolution and Applications*, 2008, vol. 2008, pp. 1–10.
17. Whitley D., Rana S., Heckendorn R. B. The Island Model Genetic Algorithm: On Separability, Population Size and Convergence, *Journal of Computing and Information Technology*, 1998, vol. 7, pp. 33–47.
18. Zhang Q., Zhou Q., Zhao S., Suganthan P. N., Liu W., Tiwari S. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, 2009.
19. Zhang Q., Liu W., Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in *Congress on Evolutionary Computation (CEC 2009)*, Trondheim, Norway, 2009, pp. 203–208.
20. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization. Barcelona, Spain, International Center for Numerical Methods in Engineering (CIMNE), 2002.
21. Berezkin V. E., Kamenev G. K., Lotov A. V. Hybrid adaptive methods for approximating a nonconvex multidimensional Pareto frontier, *Computational Mathematics and Mathematical Physics*, 2006, vol. 46, no. 11, pp. 1918–1931.
22. Zinchenko L. A., Kureichik V. M. Multialternative genetic algorithms for the search of a function extremum, *Journal of Computer and Systems Sciences International*, 2003, vol. 42, no. 4, pp. 575–583.
23. Knysh D. S., Kureichik V. M. Parallel genetic algorithms: A survey and problem state of the art, *Journal of Computer and Systems Sciences International*, 2010, vol. 49, no. 4, pp. 579–589.
24. Lunin D. V., Skvortsov S. V. Razrabotka paralelnogo geneticheskogo algoritma dlja reshenija zadachi kommivojzhhjora na platforme CUDA, *Sistemy upravlenija i informacionnye tekhnologii*, 2015, vol. 60, no. 2, pp. 50–55.
25. Mokshin V. V., Yakimov I. M. Metod formirovanija modeli analiza slozhnoj sistemy, *Informacionnye Tehnologii*, 2011, no. 5, pp. 46–51.
26. Khivintcev M. A., Akopov A. S. Primenenie mnogo-agentnogo geneticheskogo algoritma dlja poiska optimal'nyh strategicheskij i operativnyh reshenij, *Biznes-informatika*, 2014, no. 1 (27), pp. 23–33.

С. В. Зыкин, д-р техн. наук, проф., e-mail: szykin@mail.ru,
Институт математики им. С. Л. Соболева СО РАН,
В. С. Зыкин, аспирант, e-mail: vszykin@mail.ru,
Омский государственный технический университет

Коммутативные преобразования в базе данных при редактировании многотабличных запросов

Рассматриваются теоретические аспекты формирования приложений для редактирования данных в базах данных реляционного типа. Эта актуальная проблема до сих пор не получила удовлетворительного решения. Приложение должно работать с представлением данных — таблицей, сформированной из множества отношений базы данных с использованием запросов типа "соединение—селекция—проекция". Требованиям к таким приложениям является реализация функций редактирования данных: дополнение, удаление и модификация кортежей в таблице. В соответствии с выполненным редактированием приложение должно преобразовать исходную базу данных так, чтобы преобразование удовлетворяло условию коммутативности. Для указанных преобразований получены формулы реляционной алгебры, доказана корректность преобразований.

Ключевые слова: реляционная база данных, коммутативные преобразования, реляционная алгебра, редактирование представлений

Введение

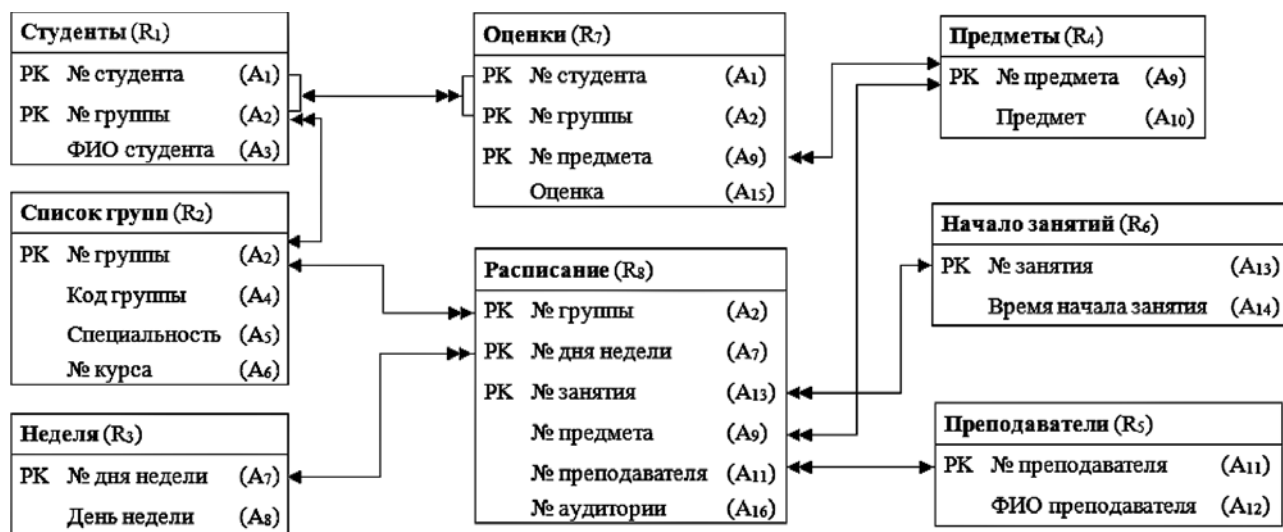
Представления (*view*) в базах данных (БД) играют важную роль как средство структурирования информации для обеспечения потребностей конкретных пользователей. Доступ к чтению через представления легко реализуется средствами языков запросов, тогда как обновление данных в БД с помощью представлений является довольно сложной проблемой. Для разработчиков программного обеспечения — систем управления базами данных (СУБД) — редактирование данных с помощью представлений является актуальной проблемой, поскольку соответствующий инструментарий существенно сократит время разработки типовых приложений.

Ограничения на обновления представлений данных присутствуют, начиная с первого стандарта языка SQL (Structured Query Language). Суть их сводится [1] к необходимости СУБД "для каждой строки представления найти соответствующую строку в исходном отношении, а для каждого обновляемого столбца представления — соответствующий столбец в исходной таблице (отношении)". При этом, "если представление пропускает столбец исходной таблицы (отношения) с ограничением NOT NULL без спецификации DEFAULT, то нельзя выполнить вставку но-

вой строки в таблицу (отношение) посредством данного представления. Причина в том, что в этом случае нет способа указать значение данных для пропущенного столбца".

Большинство коммерческих СУБД лишь частично ослабляют базовые ограничения стандарта SQL [2–6]. Например, выделенные столбцы в представлении могут быть предназначены только для просмотра, попытка редактирования таких столбцов блокируется, тогда как значения в других столбцах можно изменять в пределах заданных ограничений на соответствующие столбцы в исходном отношении.

Наиболее продвинутой технологией редактирования многотабличных представлений реализована в Microsoft Access. В режиме конструктора формируется многотабличный запрос (представление), в котором действительно допускаются операции дополнения, удаления и модификации кортежей. Однако имеются существенные недостатки. Например, в запросе требуется присутствие ключевых атрибутов для связи между отношениями. Обычно эти атрибуты — номера объектов, которые лишены семантики, что создает проблемы при вводе новых значений. Имеются очевидные ошибки, например, в представлении "Расписание занятий по специальности" (пример 1) попытка переноса одного занятия с по-



Фрагмент схемы БД "Вуз"

недельника на среду путем редактирования дня недели привела к замене всех значений "Понедельник" на значение "Среда" в справочном отношении R_3 (см. рисунок). При переходе из режима конструктора в режим SQL возможность редактирования запросов блокируется. Основная причина перечисленных и иных проблем: должно быть установлено соответствие кортежей редактируемой таблицы и кортежей отношений БД, что не всегда удается сделать. Для решения проблем редактирования представлений данных СУБД имеют механизм триггеров, но это позволяет решить только частную проблему частным способом.

Далее под представлением будем понимать таблицу данных, сформированную в результате выполнения запроса к БД. Таблицы исходной БД будем называть отношениями. Для демонстрации предлагаемых решений рассмотрим фрагмент схемы БД "Вуз" (см. рисунок). На схеме атрибуты первичного ключа отношения помечены символами РК, связи на схеме соответствуют ссылочным ограничениям целостности.

Именам отношений и атрибутов на схеме БД "Вуз" в скобках поставлены в соответствие символные обозначения.

1. Связанные работы

Работа [7] была одной из первых по данной тематике. В ней исследуются свойства отображений состояний базы данных в состояния представлений, основанных на морфизме. Представлена схема коммутативности композиции преобразований, исследования сопровождаются примерами. Однако отсутствуют какие-либо ал-

горитмы, соответствующие преобразованиям, и метод "константного дополнения", предложенный авторами [7], слишком ограничительный, чтобы быть полезным для создания технологии.

В работе [8] рассмотрена проблема автоматического перевода обновлений пользовательских представлений в соответствующие обновления исходной БД. Прежде всего, были исследованы свойства корректности таких преобразований и условия их существования. Условия были получены сначала с точки зрения расширения схемы и представления, а затем синтаксически с использованием функциональных зависимостей и ключей. Для определения операций преобразования БД были определены два графа: граф-трассировка и граф зависимостей представления. Проведенные исследования показали, что точные и уникальные переводы обновлений могут быть возможны только в довольно жестких условиях. Необходимость построения графов и условия корректности в виде исчисления предикатов делают затруднительным использование этого материала для формирования технологии. Кроме того, преобразования, описанные в работе [8], современные СУБД частично делают автоматически.

В работе [9] предпринята попытка решения данной проблемы в общем виде: допускается модификация кортежей во всех отношениях представления и работа с комбинациями представлений. Для решения проблемы неоднозначности интерпретации операций преобразования предлагается ряд эвристических критериев и правил, которые в общем случае не гарантируют корректного результата. В крайнем случае предполагается вмешательство пользователя в процесс модификации БД.

В работе [10] рассматривается обратная задача — обновление представлений данных. В общем случае обновление представлений неоднозначно и требует дополнительного представления, которое содержит всю информацию, опущенную из данного представления, и остается постоянным во время преобразований. Изучаются некоторые вычислительные проблемы, связанные с применением этой общей методологии в контексте реляционных баз данных. Показано, что поиск минимального дополнения к данному представлению является *NP*-полной задачей. Корректность преобразований проверяется тестированием, исследуется вычислительная сложность тестирования.

В работе [11] рассмотрена проблема перевода обновлений представлений в базы данных, где база данных и представление моделируются как абстракции данных. Абстракция данных состоит из набора состояний и набора примитивных операторов обновления, представляющих функции перехода состояния. Показано, как сложные программы обновления могут быть созданы из примитивных операторов обновления, и как программы обновления версий транслируются в программы обновления баз. Исследованы свойства эквивалентности и сериализуемости переходов между состояниями. В работе определено понятие дополнения и показано влияние постоянного дополнения на изменения в представлении.

В работе [12] изучается проблема трансляции обновления представления для реляционной модели данных, в которой базовые отношения могут содержать значения *null*. Предполагается наличие репрезентативного экземпляра, который считается правильным представлением всех данных в базе данных; класс изучаемых представлений состоит из полных проекций репрезентативного экземпляра. Рассматриваются только отдельные схемы баз данных, т. е. схемы, для которых глобальная согласованность подразумевается локальной согласованностью. Условие независимости отношений является сильно ограничительным. Так, на приведенном выше рисунке отношение "Расписание" содержит суперключ почти для всех отношений, следовательно, все они зависимы. Кроме того, для обеспечения корректности преобразования предполагается формирование расширения модифицируемого отношения до размеров универсума (отношения, определенного на всем множестве атрибутов), что само по себе является препятствием для практического использования такого подхода.

В работе [13] продолжены исследования свойств модификации представлений, основанных на "*constant complement approach*" [7]. Получены интересные теоретические результаты, в том числе применительно к множеству внешних схем.

Однако результаты пока далеки от практического использования.

Наиболее близка к нашей тематике работа [14], в которой обсуждается проблема корректности формирования представлений, основанных на реляционных операциях: соединение, селекция и проекция. Автором рассмотрена схема межмодельных преобразований, основанная на условиях коммутативности. Также рассмотрены пять критериев правильности выполненных преобразований, представлены варианты алгоритмов выполнения операций дополнения, удаления и модификации кортежей в алгоритмах. К недостаткам работы [14] можно отнести отсутствие модели для описания представлений, что является препятствием к созданию технологической цепочки преобразований. Рассмотренные примеры с командами SQL не заменяют отсутствующую модель. Кроме того, для обеспечения корректности преобразования требуется наличие ключевых атрибутов в представлениях. В реальных БД ключами чаще всего являются автоинкрементные атрибуты, в которых хранятся номера соответствующих объектов. Значения таких атрибутов недоступны для редактирования и не обладают какой-либо семантикой. Следовательно, работа пользователей с представлениями будет существенно затруднена.

В данной работе учтены перечисленные недостатки. Исследования доведены до уровня технологии.

2. Метод коммутативных преобразований данных

Построение межмодельных отображений начинается с выбора либо с формирования некоторой универсальной модели, которая имеет расширенные возможности в части описания и представления данных. В данной работе предполагается, что универсальная модель известна и полностью определена (реляционная). Далее формируется либо выбирается целевая (специализированная) модель данных, с которой и предполагается дальнейшая работа. Чаще всего целевая модель определяется посредством пользовательского описания данных. Конечной целью построения отображения является разработка программного обеспечения, поддерживающего интерфейс между исполняющей средой (программным обеспечением) для универсальной модели и исполняющей средой для целевой модели, в том числе для модели пользовательского представления. Существенной деталью построения является то, что пользователь должен иметь возможность корректно выполнять весь спектр операций над данными, не выходя за рамки своего приложения.

Модель информационной системы (модель данных) запишем в виде алгебраической системы:

$$\Omega = \langle M, D, O, P \rangle,$$

где M — логическая схема данных; D — совокупность допустимых состояний БД; O — набор операций для модели Ω ; P — совокупность предикатов, ограничивающих допустимые состояния D ; M и D в совокупности являются носителем системы.

Модель Ω будем считать исходной, а $\Omega' = \langle M', D', O', P' \rangle$ — целевой (пользовательской). Следовательно, необходимо построение преобразования $\Omega \Rightarrow \Omega'$.

В работе [15] рассмотрен метод построения отображения моделей данных, основанный на свойстве коммутативности. В работе [16] получено обобщение метода на случай отсутствия биективности состояний исходной и целевой моделей данных. Такая ситуация является актуальной, когда пользовательское приложение взаимодействует не со всей БД, а только с ее частью. Спецификой данной работы является полностью обновляемое представление данных пользователя при активизации приложения: выполнение многотабличного запроса. Далее рассмотрим последовательность построения преобразования исходной модели Ω в целевую Ω' .

1. Формирование многотабличного запроса осуществляется в терминах стандартного языка запросов. В настоящее время таковым является язык SQL. Запрос будет записан компактнее, если воспользоваться реляционной алгеброй [17, 18], которая является теоретической основой языка SQL. Переход из одного состояния БД в другое осуществляется за счет выполнения операции редактирования: дополнение, удаление и модификация кортежа в каком-либо отношении.

Определение 1. Совокупность всех значений данных в БД, неизменных в течение некоторого промежутка времени, будем называть элементарным состоянием.

При работе с моделью Ω' пользователь выполняет команду f'_p , которая переводит модель Ω' из состояния $d'_i \in D'$ в состояние $d'_j \in D'$. В программном обеспечении должны быть реализованы алгоритмы, выполняющие соответствующие преобразования модели Ω из состояния $d_i \in D$ в состояние $d_j \in D$.

Определение 2. Преобразование $\Omega \Rightarrow \Omega'$ будем считать корректным, если выполнены условия коммутативности:

$$d_i - (Q) \rightarrow d'_i - (f'_p) \rightarrow d'_j,$$

$$d_i - (Alg_p) \rightarrow d'_i - (Q) \rightarrow d'_j,$$

где Q — многотабличный запрос; Alg_p — алгоритм преобразования исходной БД, соответствующий команде редактирования f'_p .

Другими словами, в состояние d'_j можно перейти двумя различными способами, но результат должен быть один и тот же. В определении 2 предполагается, что при выполнении преобразований состояние БД не изменяется другими приложениями.

3. Для исходной модели данных Ω может быть задана совокупность ограничений на допустимые состояния. Для реляционной модели данных это, прежде всего, первичные ключи, внешние ключи и ограничения домена для атрибутов. Первичные ключи позволяют реализовать в БД функциональные зависимости [19]. Кроме того, функциональные зависимости могут быть реализованы с помощью уникального индекса и, в крайнем случае, за счет триггера. Теоретической основой для внешних ключей являются зависимости включения [20]. Эти зависимости реализуются за счет связей на схеме БД. Таким образом, ограничения целостности в совокупности определяют допустимые состояния данных и переходы между ними для исходной модели Ω . Поскольку запрос Q определяет соответствующие состояния моделей Ω и Ω' , то соответствующие ограничения на допустимые состояния и переходы модели Ω' также определяются этим запросом. Образы ограничений целостности должны быть достаточными для выполнения исходных ограничений модели Ω : для целевой модели Ω' могут быть заданы более "жесткие" ограничения, если реализация образов ограничений, являющихся необходимыми и достаточными, для модели Ω' является затруднительной либо невозможной. В крайнем случае ограничением целостности может стать запрет на модификацию определенной группы данных.

4. Выполнение операции преобразования f'_p для модели Ω' должно сопровождаться преобразованием состояния модели Ω . Для этого в системе должен быть реализован алгоритм Alg_p , соответствующий операции f'_p и удовлетворяющий условию коммутативности. Таким образом, для каждой операции f'_p должен быть реализован свой алгоритм преобразования БД — Alg_p . Из этого следует, что набор операций f'_p должен быть строго регламентирован.

При разработке алгоритмов Alg_p основное внимание должно быть уделено корректности преобразований. Однако выполнение алгоритмов Alg_p предполагается при работе пользователя с приложением (*online*), следовательно, должно быть обеспечено приемлемое время их выполнения.

3. Редактирование многотабличных представлений данных

Рассмотрим формальную постановку задачи. Исходная реляционная база данных (РБД) пред-

ставлена в виде множества отношений R_1, R_2, \dots, R_k . Целевая модель — реляционная таблица данных, являющаяся результатом выполнения запроса к РБД:

$$Q = \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m])),$$

где π_{X_0} — операция проекции по множеству атрибутов X_0 ; σ_F — операция селекции; F — логическое выражение на атрибутах отношений R'_1, R'_2, \dots, R'_m ; \bowtie — операция естественного соединения отношений РБД; $R'_i[X_i]$ — краткая запись операции проекции отношения R'_i по атрибутам X_i . Атрибут A_j отношения R'_i принадлежит X_i , $i = 1, 2, \dots, m$: если $A_j \in X_0$, и/или существует R'_l , $l \neq i$: $A_j \in \langle R'_l \rangle$, и/или $A_j \in \langle F \rangle$, других атрибутов в X_i нет.

Рассмотрим соответствие обозначений. Схема данных исходной модели данных есть множество схем отношений БД: $M = \{\langle R_1 \rangle, \langle R_2 \rangle, \dots, \langle R_k \rangle\}$, где $\langle R_i \rangle$ — все атрибуты отношения R_i ; $M' = \bigcup_{i=1}^m \langle R'_i \rangle$ — заголовок таблицы, соответствующий результату выполнения запроса Q ; P — первичные и внешние ключи БД; P' — реализованные зависимости БД (образы первичных и внешних ключей); O и O' — базисные для БД операции дополнения, удаления и модификации кортежей в отношениях. Допустимые состояния D и D' определяются предикатами P и P' соответственно.

Условие коммутативности для запроса интерпретируется следующим образом: если пользователь удалил, добавил или модифицировал кортеж, то после повторной загрузки данных именно этот кортеж должен быть удален, добавлен или модифицирован. Остальные кортежи должны остаться без изменений с точностью до порядка их следования.

Рассмотрим требования к составу и структуре запроса Q .

1. Отношения R'_1, R'_2, \dots, R'_m должны иметь упорядочение по внешним ключам: главные отношения в последовательности стоят раньше, подчиненные — позже. Таким образом, должен существовать частичный порядок, в котором есть только одно отношение R'_m , не имеющее подчиненных отношений. Это отношение будет соответствовать семантике приложения: выполненные операции в приложении будут реализовываться только соответствующими операциями в R'_m . Далее это отношение будем называть целевым. Фактическое установление частичного порядка над отношениями не является принципиальным, однако алгоритмическая реализация преобразования БД в виде схемы итерирования отношений [21] в большинстве случаев позволяет

существенно сократить промежуточные результаты за счет внешних ключей и, как следствие, добиться сокращения времени работы алгоритма.

Сформулируем условия (ограничения) на атрибуты запроса Q . Для корректного выполнения операций должны быть выполнены следующие условия.

2. $\langle R'_m \rangle \rightarrow X_0$: атрибуты отношения R'_m функционально определяют атрибуты приложения X_0 .

3. $\langle R'_m \rangle \subseteq X_0 \cup X_1 \cup \dots \cup X_{m-1}$ — отсутствие свободных атрибутов в целевом отношении.

Представленные условия далее должны быть реализованы в инструментальной исполняющей среде при генерации пользовательских многотабличных приложений. С использованием фрагмента БД (см. рисунок) могут быть сгенерированы представления для различных приложений: "Ведомость на экзамен", "Сводная ведомость по предмету", "Сводная ведомость для группы", "Расписание занятий для группы", "Расписание занятий на день недели" и т. п. Каждый кортеж представления для перечисленных приложений имеет ровно один кортеж прообраз в целевом отношении, что упрощает процесс редактирования. Рассмотрим более сложный пример приложения "Расписание занятий по специальности", в котором одному кортежу представления соответствует несколько кортежей целевого отношения.

Пример 1. С использованием обозначений, данных на приведенном выше рисунке, запрос на расписание занятий по специальности может иметь следующий вид:

$$Q_1 = \pi_{X_0}(\sigma_{A_{10}=\text{Предмет}}(R_2[A_2 A_3 A_6] \bowtie R_3 \bowtie R_4 \bowtie R_5 \bowtie R_6 \bowtie R_8)),$$

где $X_0 = A_5 A_6 A_8 A_{14} A_{12} A_{16}$. При старте приложения значение параметра 'Предмет' выбирается из списка предметов (таблица R_4). С помощью приложения, в основе которого используется представление Q_1 , должна быть возможность дополнения и удаления занятия. Кроме того, должны быть доступны: перенос занятия на другое время и/или день недели, замена преподавателя, замена аудитории.

3.1. Операция удаления кортежа

Пусть в приложении кортеж u удален из таблицы, соответствующей запросу Q . Далее запрос и результат его выполнения будем обозначать одним и тем же символом Q .

В соединении отношений

$$R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m]$$

множество кортежей T , удовлетворяющих логическому выражению F и принимающих значения

кортежа u на атрибутах X_0 , можно выразить формулой

$$T = \sigma_{F \wedge (X_0=u)} (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m]),$$

где выражение $(X_0 = u)$ означает равенство значений одноименных атрибутов. Тогда операция удаления кортежа u из таблицы Q сводится к удалению кортежей

$$R''_m = R'_m \setminus \pi_{\langle R'_m \rangle} (T) \quad (1)$$

в отношении R'_m , где символом \setminus помечена реляционная операция вычитания [18]. Покажем, что данное преобразование будет корректным (коммутативным). Это означает, что таблица

$$Q' = \pi_{X_0} (\sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R''_m [X_m]))$$

будет отличаться от таблицы Q только отсутствием одного кортежа u . Заметим, что при выполнении операции проекции дублированные кортежи удаляются.

Теорема 1. Операция удаления кортежа $u \in Q$ коммутативна.

Доказательство. 1. Покажем, что кортеж u будет отсутствовать в Q' . Допустим обратное: $u \in Q'$. Следовательно, существует кортеж $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R''_m [X_m])$ и $t[X_0] = u$. Поскольку $R''_m [X_m] \subseteq R'_m [X_m]$, то $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$ и, следовательно, должен быть удален, что противоречит предположению.

2. Покажем, что не произошло удаление лишних кортежей, если $u' \in Q$ и $u' \neq u$, тогда $u' \in Q'$. Предположим обратное: кортеж u' отсутствует в Q' . Следовательно, любой кортеж $t' \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$, для которого $t'[X_0] = u'$, должен удовлетворять условию для удаления $t'[\langle R'_m \rangle] \in T$. Однако в T содержатся только те кортежи, для которых существует $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$ и $t[X_0] = u$. Следовательно, для любого кортежа t' найдется кортеж t , для которых $t'[\langle R'_m \rangle] = t[\langle R'_m \rangle]$ и $t'[X_0] \neq t[X_0]$, что противоречит ограничению $\langle R'_m \rangle \rightarrow X_0$. Полученные противоречия доказывают коммутативность операции удаления кортежа.

Аналитическое выражение (1) является замещением алгоритма Alg_1 для удаления кортежа в многотабличном запросе. В исполняющей среде это выражение достаточно для формирования команды SQL с последующим ее выполнением посредством какой-либо СУБД. Однако эффективное выполнение этой команды не гарантируется, в том числе по причине значительного объема промежуточных данных. В исполняющей среде

предлагается реализовать алгоритм итерирования отношений с динамическим выбором последовательности просматриваемых отношений [21].

3.2. Операция дополнения кортежа

По аналогии с предыдущей операцией рассмотрим аналитическое выражение операции дополнения кортежа в терминах реляционной алгебры. Пусть кортеж u дополняется к таблице Q , тогда в исходной БД должны быть выполнены следующие преобразования:

$$R''_m = R'_m \cup T, \quad (2)$$

где

$$T = \pi_{X_m} (\sigma_F (T' \bowtie u));$$

$$T' = \pi_Y (\sigma_{Z=u[Z] \& F} (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_{m-1} [X_{m-1}]));$$

$$Z = X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1});$$

$$Y = (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}),$$

если $Z = \emptyset$, то будем считать, что формула $Z = u[Z]$ принимает значение "Истина", F — проекция формулы F на подпространство атрибутов $\langle F \rangle = \langle F \rangle \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1})$ [22]. В случае операции дополнения решение может отсутствовать по следующим причинам: 1) может оказаться пустым множество T' — отсутствуют кортежи, с которыми может быть выполнено соединение (формула $Z = u[Z]$ принимает значение "Ложь" для всех кортежей в T'); 2) может оказаться пустым множество T — отсутствуют кортежи, на которых F принимает значение "Истина".

Покажем, что данное преобразование будет корректным (коммутативным), если $T' \neq \emptyset$ и $T \neq \emptyset$. Это означает, что таблица

$$Q' = \pi_{X_0} (\sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_{m-1} [X_{m-1}] \bowtie R''_m [X_m]))$$

будет отличаться от таблицы Q только присутствием одного кортежа u .

Теорема 2. Если $T' \neq \emptyset$ и $T \neq \emptyset$, то операция дополнения кортежа $u \in Q$ коммутативна.

Доказательство. Все кортежи Q будут принадлежать Q' , поскольку в отношениях R'_i , $i = 1..m$, кортежи не были удалены. Покажем, что кортеж u , и только он, будет добавлен в Q' . Предварительно заметим, что $\langle R'_m \rangle = X_m$, т. е. все атрибуты целевого отношения R'_m участвуют в реализации запросов Q и Q' . Допустим обратное, существует атрибут $A_j \in \langle R'_m \rangle$ и $A_j \notin X_m$. Тогда по правилам формирования X_i должно быть выполнено $A_j \notin X_i$, $i = 1, \dots, m-1$. Из ограничения 3 (см. разд. 3) следует, что $A_j \in X_0$, но такие атрибуты должны быть включены в X_m по построению. Следова-

тельно, $\langle R'_m \rangle \subseteq X_m$. Включение в обратную сторону следует из свойства операции проекции. Таким образом, имеем

$$X_m \subseteq X_0 \cup X_1 \cup X_2 \cup \dots \cup X_{m-1}$$

и

$$X_0 \subseteq X_1 \cup X_2 \cup X_3 \cup \dots \cup X_m.$$

С учетом полученных неравенств все кортежи T на атрибутах Z совпадают с кортежем $u[Z]$ по построению. При формировании T кортежи по атрибутам $X_0 \setminus Z$ будут равны значениям соответствующих атрибутов $u[X_0 \setminus Z]$. По свойствам операций реляционной алгебры имеем:

$$\begin{aligned} Q' &= \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie R'_m[X_m])) = \\ &= \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie (R'_m[X_m] \cup T))) = \\ &= Q \cup \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie T)). \end{aligned}$$

Множество кортежей $\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie T)$ по предположению не пусто, и все кортежи этого множества по атрибутам X_0 совпадают с кортежем u . Следовательно, таблица Q' будет отличаться от Q только наличием одного и только одного кортежа u . Что и требовалось доказать.

Замечание. Для выполнения условия коммутативности в формуле (2) к R'_m достаточно добавить только один кортеж из множества T , однако необходимо дополнение всех кортежей множества T , что соединит вновь введенную информацию со всей существующей информацией в БД. Это обеспечивается выбором значений свободных атрибутов $X_1 \cup X_2 \cup X_3 \cup \dots \cup X_m \setminus Z$ при формировании T .

Как и в случае с операцией удаления кортежа, полученное аналитическое выражение (2) является замещением алгоритма Alg_2 для дополнения кортежа в многотабличном запросе. В исполняющей среде формируется команда SQL, соответствующая формуле (2).

3.3. Операция модификации кортежа

В общем случае операция модификации кортежа сводится к операциям удаления старого кортежа u и дополнения нового u' . Алгоритму Alg_3 (операция модификации) соответствует следующее выражение реляционной алгебры:

$$\begin{aligned} R''_m &= R'_m \setminus \pi_{\langle R'_m \rangle}(\sigma_{F \wedge (X_0=u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \\ &\dots \bowtie R'_m[X_m])) \cup \pi_{X_m}(\sigma_F(\pi_Y(\sigma_{Z=u[Z] \& F}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \\ &\dots \bowtie R'_{m-1}[X_{m-1}]) \bowtie u')), \end{aligned}$$

где

$$\begin{aligned} Z &= X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}); \\ Y &= (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}). \end{aligned}$$

Применив эквивалентные преобразования, получим:

$$R''_m = R'_m \setminus \pi_{X_m}(\sigma_F(T \bowtie u)) \cup \pi_{X_m}(\sigma_F(T \bowtie u')), \quad (3)$$

где

$$T = \pi_Y(\sigma_{(Z=u[Z]) \vee (Z=u'[Z]) \& F}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}]));$$

$$\begin{aligned} Z &= X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}); \\ Y &= (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}). \end{aligned}$$

В формуле (3) использованы обозначения формулы (2) и следующие преобразования. Выражение $\sigma_{F \wedge (X_0=u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m])$ было заменено на эквивалентное $\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie u)$. В T объединены списки удаляемых и дополняемых кортежей. Если какие-либо кортежи в T принадлежат обоим спискам, то при формировании $T \bowtie u$ и $T \bowtie u'$ каждый кортеж участвует только в своей операции и лишние кортежи в T игнорируются. Заметим, если $u' = \emptyset$, то формула (3) эквивалентна операции удаления кортежа u , если $u = \emptyset$, то (3) — дополнение кортежа u' .

В большинстве приложений каждый кортеж-представление имеет один кортеж-прообраз, в целевом отношении, также достаточно выполнить одну команду UPDATE.

Пример 2. Для демонстрации рассмотрим операцию модификации кортежа в примере 1. Пусть при старте приложения выбрано ограничение на название предмета "Введение в специальность". Схема представления (заголовок) следующая: $\langle \text{Специальность}(A_5) \rangle$, $\langle \text{№ курса}(A_6) \rangle$, $\langle \text{День недели}(A_8) \rangle$, $\langle \text{Время начала занятия}(A_{14}) \rangle$, $\langle \text{ФИО преподавателя}(A_{12}) \rangle$, $\langle \text{№ аудитории}(A_{16}) \rangle$. Пусть в представлении кортеж (Физика, 1, Вторник, 11-40, Иванов И. И., Б-314) заменили кортежем (Химия, 1, Среда, 9-45, Петров П. П., А-212). Исходному кортежу в целевом отношении будет соответствовать три кортежа (три группы на специальности 'Физика'): $(A_2 = 15, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$, $(A_2 = 16, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$, $(A_2 = 17, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$. В соответствии с предложенной технологией в целевом отношении перечисленные корте-

жи должны быть заменены четырьмя кортежами (четыре группы на специальности 'Химия'): ($A_2 = 22, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$), ($A_2 = 23, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$), ($A_2 = 24, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$), ($A_2 = 25, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$). Если бы значения атрибута A_2 совпадали для всех новых кортежей, было бы достаточно одной команды UPDATE для выполнения замены. Однако для данного примера самым эффективным способом является выполнение двух команд SQL: удаление (DELETE) старых кортежей и дополнение (INSERT) новых.

Заключение

Рассмотренная в данной работе теория является основой для создания новой технологии генерации многотабличных приложений. Для этого создается программа-шаблон, в которую в диалоговом режиме вставляются компоненты запроса Q . В ходе диалога проверяются ограничения на запрос с использованием реализованных зависимостей. Операции реляционной алгебры в виде команд SQL вставляются в программу-шаблон. В результате получается готовая к исполнению программа, реализующая редактирование многотабличных представлений. Аналогичная по структуре технология была предложена в работе [23] и реализована для динамического формирования многомерных данных (плоский вариант) в среде Microsoft Office [24].

Список литературы

1. **Forta B.** SQL in 10 Minutes, Sams Teach Yourself. 4th Ed. Indiana, USA: Sams Publishing PTG, 2013. 288 p.
2. **Baklarz G.** DB2 9 for Linux, UNIX, and Windows: DBA Guide 6 edition. NY, USA: IBM Press, 2007. 1136 p.
3. **Oracle** Database Online Documentation 11g Release 1 (11.1). Database Administrator's Guide, 2017. 299 p. URL: https://docs.oracle.com/cd/B28359_01/server.111/b28310/toc.htm (дата обращения: 31.10.17).
4. **Бондарь А.** Microsoft SQL Server 2014. СПб: БХВ-Петербург, 2015. 592 с.
5. **McLaughlin B.** PHP & MySQL: The Missing Manual, 2nd Edition. Sebastopol, USA: O'Reilly Media, 2012. 546 p.
6. **Worsley J., Drake J.** Practical PostgreSQL. Sebastopol, USA: O'Reilly Media, 2011. 640 p.
7. **Bancilhon F., Spyrtatos N.** Update semantics of relational views // ACM Trans. Database Syst. 1981. Vol. 6, N. 4. P. 557–575.

8. **Dayal U., Bernstein P. A.** On the correct translation of update operations on relational views // ACM Trans. Database Syst. 1982. Vol. 7, N. 3. P. 381–416.
9. **Masunaga Y.** A Relational Database View Update Translation Mechanism // In Proceedings of the 10th International Conference on Very Large Data Bases (VLDB '84), San Francisco, CA, USA. 1984. P. 309–320.
10. **Cosmadakis S., Papadimitriou C.** Updates of Relational Views // J. ACM. 1984. Vol. 31, N. 4. P. 742–760.
11. **Gottlob G., Paolini P., Zicari R.** Properties and update semantics of consistent views // ACM Trans. Database Syst. 1988. Vol. 13, N. 4. P. 386–524.
12. **Langerak R.** View updates in relational databases with an independent scheme // ACM Trans. Database Syst. 1990. Vol. 15, N. 1. P. 40–66.
13. **Lechtenböcker J.** The impact of the constant completion approach towards view updating // In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '03). ACM, New York, NY, USA. 2003. P. 49–55.
14. **Keller A.** Algorithms for translating view updates to database updates for views involving selections, projections, and joins // In Proceedings of the fourth ACM SIGACT-SIGMOD symposium on Principles of database systems (PODS '85). ACM, New York, NY, USA. 1985. P. 154–163.
15. **Калиниченко Л. А.** Методы и средства интеграции неоднородных баз данных. М.: Наука, 1983. 423 с.
16. **Zykin S. V.** Generation of User View for a Relational Database by Mappings // Programming and Computer Software. 1999. Vol. 25, N. 3. P. 173–183.
17. **Maier D.** The theory of relational databases // Computer Science Press, 1983. 637 p.
18. **Ullman J.** Principles of database systems // Computer Science Press, 1980. 379 p.
19. **Hartmann S., Link S.** The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations // ACM Transactions on Database Systems. 2012. Vol. 37, N. 2. P. 1–40.
20. **Köhler H., Link S.** Inclusion Dependencies Reloaded // The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), Melbourne, Australia, October 18–23, 2015. P. 1361–1370.
21. **Зыкин С. В., Полуянов А. Н.** Итерирование отношений базы данных для запросов специального вида // Материалы X Международной IEEE научно-технической конференции "Динамика систем, механизмов и машин", Омск, 15–17 ноября, 2016. Т. 4, № 1. С. 18–21.
22. **Mosin S. V., Zykin S. V.** Truth Space Method for Caching Database Queries // Моделирование и анализ информационных систем. 2015. Т. 22, № 2. С. 248–258.
23. **Zykin S. V.** Automation of the Interface Formation Between Multidimensional and Relational Representation of the Data // Relational Databases and Open Source Software Developments. NY: Nova Science Publishers, 2011. P. 43–66.
24. **Зыкин С. В.** Автоматизация формирования табличных приложений // Моделирование и анализ информационных систем. 2013. Т. 20, № 4. С. 41–54.

S. V. Zykin, D. Sc., Professor, Head of Laboratory, e-mail: szykin@mail.ru,
Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia,
V. S. Zykin, Postgraduate, e-mail: vszykin@mail.ru,
Omsk State Technical University, Omsk, Russia

Commutative Conversion in the Database when Editing a Multitable Query

Theoretical aspects of the formation of applications for editing data in relational databases are considered. This important problem has not yet been satisfactorily resolved. That problem has been solved by "hard" restrictions on the views and it requires the highly qualified user when using these components. In this paper, the theoretical aspects of the applications formation for work with a relational database are discussed. The application works with a data view such as table formed by a set of database relations using "join-select-projection" queries. The implementation of data editing functions: inserting, deleting and updating of tuples in the table are the requirement for such applications. The application must convert the source database according to the user's view editing providing that the conversion satisfy the commutativity condition. Formulas of relational algebra are obtained for these transformations. The correctness of the transformations is proved. The theory in this paper is the basis for creating a new technology for generating of multitable applications. The technology involves creating a template program, where a query and its restrictions are implemented. The operations of relational algebra are transformed into SQL commands. As a result, the user obtains a ready-to-run program that implements editing of multitable views.

Keywords: relational database, commutative transformations, relational algebra, views editing

References

1. Forta B. *SQL in 10 Minutes, Sams Teach Yourself*, 4th Edition. Indiana, USA: Sams Publishing PTG, 2013. 288 p.
2. Baklarz G. *DB2 9 for Linux, UNIX, and Windows: DBA Guide* 6 edition. NY, USA, IBM Press, 2007, 1136 p.
3. Oracle Database Online Documentation 11g Release 1 (11.1). Database Administrator's Guide, 2017. 299 p. URL: https://docs.oracle.com/cd/B28359_01/server.111/b28310/toc.htm (date of access: 31.10.17).
4. Bondar' A. *Microsoft SQL Server 2014*. SPb: BHV-Peterburg, 2015. 592 p. (in Russian)
5. Maklaffin B. *PHP & MySQL: The Missing Manual*, 2nd Edition. Sebastopol, USA, O'Reilly Media, 2012, 546 p.
6. Worsley J., Drake J. *Practical PostgreSQL*, Sebastopol, USA, O'Reilly Media, 2011, 640 p.
7. Bancilhon F., Spyratos N. Update semantics of relational views, *ACM Trans. Database Syst.*, 1981, vol. 6, no. 4, pp. 557–575.
8. Dayal U., Bernstein P. A. On the correct translation of update operations on relational views, *ACM Trans. Database Syst.*, 1982, vol. 7, no. 3, pp. 381–416.
9. Masunaga Y. A Relational Database View Update Translation Mechanism, *In Proceedings of the 10th International Conference on Very Large Data Bases (VLDB '84)*, San Francisco, CA, USA, 1984, pp. 309–320.
10. Cosmadakis S., Papadimitriou C. Updates of Relational Views, *J. ACM*, 1984, vol. 31, no. 4, pp. 742–760.
11. Gottlob G., Paolini P., Zicari R. Properties and update semantics of consistent views, *ACM Trans. Database Syst.*, 1988, vol. 13, no. 4, pp. 386–524.
12. Langerak R. View updates in relational databases with an independent scheme, *ACM Trans. Database Syst.*, 1990, vol. 15, no. 1, pp. 40–66.
13. Lechtenbörger J. A The impact of the constant complement approach towards view updating, *In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '03)*, ACM, New York, NY, USA, 2003. P. 49–55.
14. Keller A. Algorithms for translating view updates to database updates for views involving selections, projections, and joins, *In Proceedings of the fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS '85)*, ACM, New York, NY, USA, 1985. P. 154–163.
15. Kalinichenko L. A. *Metody i sredstva integracii neodnorodnyh baz dannyh* (Methods and facilities for integrating nonhomogeneous databases), Moscow, Nauka, 1983. 423 p. (in Russian).
16. Zykin S. V. Generation of User View for a Relational Database by Mappings, *Programming and Computer Software*, 1999, vol. 25, no. 3, pp. 173–183.
17. Maier D. *The theory of relational databases*, Computer Science Press, 1983, 637 p.
18. Ullman J. *Principles of database systems*, Computer Science Press, 1980, 379 p.
19. Hartmann S., Link S. The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations, *ACM Transactions on Database Systems*, 2012, vol. 37, no. 2, pp. 1–40.
20. Köhler H., Link S. Inclusion Dependencies Reloaded, *The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*, Melbourne, Australia, October 18–23, 2015, pp. 1361–1370.
21. Zykin S. V., Polujanov A. N. Iterirovanie otnoshenij bazy dannyh dlja zaprosov special'nogo vida (Iteration of database relations for special queries), *Materialy X Mezhdunarodnoj IEEE nauchno-tehnicheskoy konferencii "Dinamika sistem, mehanizmov i mashin"*, Omsk, 15–17 November, 2016, vol. 4, no. 1, pp. 18–21 (in Russian).
22. Mosin S. V., Zykin S. V. Truth Space Method for Caching Database Queries, *Modelirovanie i analiz informacionnyh sistem*, 2015, vol. 22, no. 2, pp. 248–258.
23. Zykin S. V. Automation of the Interface Formation Between Multidimensional and Relational Representation of the Data. *Relational Databases and Open Source Software Developments*. NY, Nova Science Publishers, 2011, pp. 43–66.
24. Zykin S. V. Avtomatizacija formirovanija tablichnyh prilozhenij (Automating the formation of tabular applications), *Modelirovanie i analiz informacionnyh sistem*, 2013, vol. 20, no. 4, pp. 41–54 (in Russian).

К. А. Щеглов, аспирант, e-mail: skd@npp-itb.spb.ru,
А. Ю. Щеглов, д-р техн. наук, проф., e-mail: info@npp-itb.spb.ru,
Университет ИТМО, Санкт-Петербург, Россия

Моделирование угроз целевых атак

Рассматривается подход к моделированию угроз целевых атак, отличающийся использованием потенциальными нарушителями при осуществлении атаки уязвимостей "нулевого дня". Эта особенность угрозы принципиально меняет задачу моделирования, поскольку в данном случае в модели уже отсутствует параметр интенсивности устранения уязвимости — она не устраняется, так как об уязвимости нулевого дня еще не известно разработчику уязвимого программного средства. Обсуждается, как учесть эту особенность угрозы атаки в математической модели, и какие количественные характеристики (мера) актуальности угрозы атаки при этом могут быть определены.

Ключевые слова: угроза атаки, целевая атака, актуальность, количественная мера, математическое моделирование

Введение

Основной проблемой современной информационной безопасности становятся целевые атаки. Это обусловлено тем, что несмотря на пока небольшой (порядка 1 %) процент от общего числа атак, нанесенный ущерб от целевых атак уже во много раз превосходит ущерб от массовых атак [1–4].

Целевые (или таргетированные) атаки — это атаки, направленные на конкретные коммерческие организации или государственные ведомства в целях получения несанкционированного доступа к информации. Такие атаки не носят массовый характер и готовятся достаточно длительный период [1–4].

В отличие от массовых атак компьютерных вирусов (цель которых — заражение максимального числа компьютеров), подобные атаки тщательно планируются, а вредоносное программное обеспечение специально разрабатывается для каждой атаки, чтобы антивирусы и иные средства защиты, используемые в информационной системе, не смогли обнаружить атаку. Как правило, для реализации целевой атаки используются уязвимости нулевого дня, не известные и не устраняемые на момент осуществления атаки.

Таким образом, основу целевой атаки составляет использование уязвимостей реализации программных средств (ошибок программирования, общая классификация угроз уязвимостей пред-

ставлена в работе [5]), о которых еще кому-либо, кроме потенциального нарушителя, не известно.

Общий подход к математическому моделированию угрозы атаки как случайного процесса возникновения и устранения реальной угрозы (все создающие ее уязвимости выявлены, но еще не устранены) изложен в работе [5]. Уточним понятие "резервирование по угрозам уязвимостей" (или угроз уязвимостей). Именно возникновение в системе уязвимости позволяет реализовать атаку, т. е. угроза атаки создается угрозой возникновения в системе уязвимости (угрозой уязвимости). В общем случае для реализации атаки злоумышленнику может потребоваться воспользоваться несколькими разнородными, в том числе возникшими в различных средствах, одновременно присутствующими в системе уязвимостями. Вероятность такого события и, как следствие, актуальность угрозы такой атаки, естественно, будут ниже. Это позволяет интерпретировать угрозу атаки схемой параллельного резервирования (горячий резерв) создающих ее угроз уязвимостей, так как отказ безопасности системы, позволяющий реализовать атаку, возникает только при условии одновременного присутствия в системе всех уязвимостей, угрозы которых создают угрозу атаки. Если хотя бы одной уязвимости не возникнет, то реализация атаки будет невозможна. При этом соответствующим образом может интерпретироваться и задача построения системы защиты — система защиты должна обеспечивать невозмож-

ность осуществления атаки, без возникновения соответствующей уязвимости и непосредственно в системе защиты. Как видим, при такой интерпретации построения системы защиты угрозой уязвимости системы защиты, без возникновения которой невозможна атака, резервируются угрозы уязвимостей защищаемой системы [5].

Рассмотрим, как можно учесть при моделировании угрозы атаки тот факт, что атакой используется уязвимость нулевого дня, устранение которой до момента успешной реализации атаки не осуществляется.

1. Моделирование угроз массовых атак

Как отмечалось выше, в работе [5] изложены и обоснованы общие подходы к моделированию угроз атак и рассмотрены вопросы моделирования угроз массовых атак. Подход к моделированию проиллюстрируем на примере угрозы атаки, создаваемой угрозами уязвимостей двух типов. При этом моделируется марковский случайный процесс с дискретными состояниями и непрерывным временем. Искомые стационарные вероятности находятся решением соответствующей системы линейных уравнений [5]. Модель приведена на рис. 1.

Здесь $\lambda_1, \mu_1, \lambda_2, \mu_2$ — соответственно параметры угроз уязвимостей первого и второго типов — их интенсивности возникновения и устранения, значения которых могут определяться с использованием соответствующей статистики.

Состояния системы обозначены S_{ij} , где S_{00} — состояние, характеризуемое отсутствием уязвимостей в системе; S_{10} — состояние, характеризуемое присутствием в системе уязвимости первого типа; S_{01} — состояние, характеризуемое присутствием в системе уязвимости второго типа; S_{11} — состояние, характеризуемое присутствием в си-

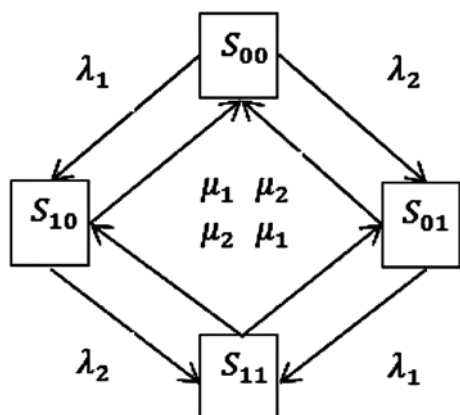


Рис. 1. Размеченный граф системы состояний случайного процесса угрозы массовой атаки

стеме уязвимости первого и уязвимости второго типов — состояние реальной угрозы атаки.

Моделируется стационарный режим, в результате моделирования определяются значения искомых параметров — интенсивности возникновения и устранения реальной (для реализации) угрозы атаки — и характеристик — стационарной вероятности готовности системы к безопасной эксплуатации, среднее время наработки системы на отказ безопасности и среднее время восстановления безопасности системы (в данном случае в отношении угрозы атаки). Таким образом, без использования каких-либо экспертных оценок может быть получена всесторонняя количественная оценка актуальности угрозы массовой атаки.

Рассмотрим особенности угрозы массовой атаки, учитываемые при построении соответствующих математических моделей. В данных моделях предполагается, что возникшая уязвимость программного средства сразу начинает устраняться его разработчиком, проще говоря, о возникшей уязвимости потенциальные нарушители (причем массово) и разработчик программного средства узнают практически одновременно. При этом потенциальный нарушитель может осуществить соответствующую атаку на информационную систему в промежутке времени с момента возникновения реальной угрозы атаки, о которой стало известно, до тех пор, пока она не будет соответствующим образом устранена, т. е. пока не будут сделаны исправления и соответствующие обновления программного средства (средств) не будут установлены в информационной системе.

2. Моделирование угроз эксклюзивных и гибридных целевых атак

Угроза атаки, относимая нами к разряду эксклюзивных целевых, характеризуется совсем иным образом. Принципиально подобная атака отличается тем, что при ее реализации используются уязвимости, известные только потенциальному нарушителю и не известные разработчику. При этом подобная угроза целевой атаки после первой же ее реализации становится угрозой массовых атак, так как об использованных при реализации атаки уязвимостях становится известно.

Принципиальным отличием предложенного подхода к моделированию эксклюзивной целевой атаки, проводимого в целях оценки ее актуальности, состоит в том, что в данном случае следует говорить о моделировании невозстанавливаемой системы — о невозстанавливаемом отказе безопасности. Это обуславливается тем, что возникающие уязвимости в программных средствах (их отказы безопасности) в данном случае

не устраняются, поскольку о них известно только потенциальному нарушителю.

Таким образом, для моделирования эксклюзивной целевой атаки может использоваться основной (или экспоненциальный) закон надежности, позволяющий в рамках теории надежности моделировать невосстанавливаемые отказы. В данном случае характеристикой угрозы атаки уже является среднее время наработки информационной системы до отказа безопасности.

Однако для получения возможности моделирования частично эксклюзивной (или гибридной) целевой атаки, о чем будет сказано далее, в данном случае также целесообразно построение соответствующей марковской модели угрозы атаки, при этом в качестве состояний системы должны использоваться невозвратные и поглощающее состояния (моделируется нестационарный режим).

Пример такой модели приведен на рис. 2, а.

Модель построена в предположении о том, что угроза атаки создается угрозами уязвимостей двух типов.

В модели, приведенной на рис. 2, а, S_{11} — это поглощающее состояние, после перехода в которое процесс завершается (стационарные вероятности для такой модели отсутствуют).

Состояния же S_{00} , S_{10} , S_{01} — невозвратные, характеризуются соответствующим средним временем пребывания в них системы T_{00} , T_{10} , T_{01} . Угроза атаки в соответствии с этой моделью невосстанавливаемой системы характеризуется средним временем наработки системы до отказа безопасности $T_{но}$, определяемым как

$$T_{но} = T_{00} + T_{10} + T_{01}.$$

Для построения искомой системы линейных уравнений можно использовать преобразование Лапласа — здесь уже необходимо определять не

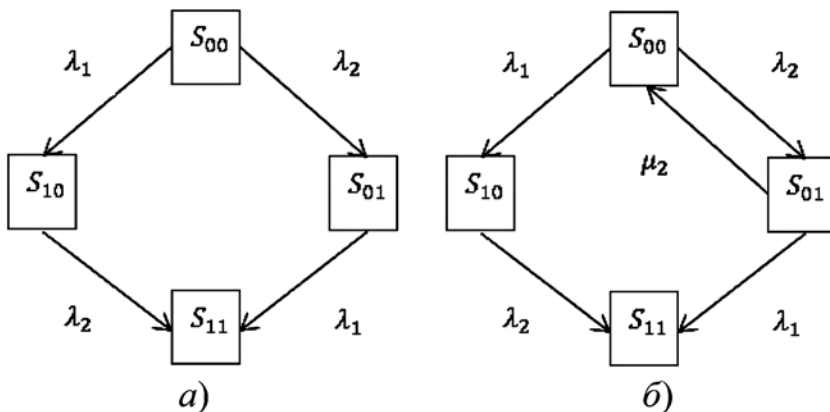


Рис. 2. Иллюстрация моделей угроз целевых атак:
а — эксклюзивная; б — гибридная

стационарные вероятности (система не стационарна), а временные характеристики — среднее время пребывания системы в соответствующих состояниях, а можно исходить из несколько иной интерпретации модели, представленной в работе [5]: n_i — это среднее число возникших уязвимостей i -го типа, переводящее систему из некоего состояния S_{rd} , определяемое как $n_i = \lambda_i T_{rd}$, где T_{rd} — среднее время пребывания системы в состоянии S_{rd} . Этими значениями взвешиваются дуги в графе переходов состояний, указывая на то, сколько в среднем переходов будет осуществлено из соответствующего состояния под воздействием какого-либо входного потока случайных событий.

При подобной интерпретации модели достаточно просто могут быть определены все искомые уравнения. Первое получаем исходя из того, что из состояния S_{00} будет осуществлен только один переход:

$$T_{00}(\lambda_1 + \lambda_2) = 1.$$

Среднее число переходов, переводящих систему в состояния S_{10} и S_{01} и переводящих систему из этих состояний, должно соответствующим образом совпадать, из чего получаем необходимые нам уравнения:

$$\begin{aligned} T_{00}\lambda_1 &= T_{01}\lambda_2; \\ T_{00}\lambda_2 &= T_{10}\lambda_1. \end{aligned}$$

Этих уравнений достаточно для получения решения. Последнее же уравнение для модели, приведенной на рис. 2, а, на основе того, что в состоянии S_{11} может быть осуществлен только один переход, имеет вид

$$T_{01}\lambda_2 + T_{10}\lambda_1 = 1.$$

Решая полученную систему линейных уравнений, определяем характеристику среднего времени наработки до отказа безопасности $T_{но}$ для модели, представленной на рис. 2, а:

$$T_{но} = \frac{1}{(\lambda_1 + \lambda_2)} \left(1 + \frac{\lambda_1}{\lambda_2} + \frac{\lambda_2}{\lambda_1} \right).$$

На практике более вероятны целевые атаки, отличающиеся тем, что потенциальному нарушителю известны уязвимости нулевого дня не для всех типов уязвимостей, создающих угрозу атаки. Назовем та-

кие угрозы атак частично эксклюзивными или гибридными. Понятно, что такая целевая атака возможна в том случае, если угроза атаки создается угрозами уязвимостей нескольких типов.

Подобную угрозу целевой атаки с использованием предложенного подхода также можно промоделировать, пример модели подобной угрозы атаки приведен на рис. 2, б. В данном случае опять же моделируется невозстанавливаемая система в целом, однако при моделировании события возникновения используемых при реализации атаки известных (устраняемых) уязвимостей (для модели, приведенной на рис. 2, б, это уязвимости второго типа) уже используются сообщающиеся состояния: S_{00}, S_{01} .

Требуемые линейные уравнения опять же при этом могут быть получены с использованием рассмотренной ранее интерпретации переходов между состояниями. Например, с учетом того, что из состояния S_{00} число переходов будет на 1 больше, чем число возвратов в это состояние, первое уравнение будет иметь следующий вид:

$$T_{00}(\lambda_1 + \lambda_2) - T_{01}\mu_2 = 1.$$

По аналогии с тем, как это было сделано ранее, можно построить и остальные требуемые уравнения.

Решение данной задачи моделирования позволяет количественно определять то, насколько изменится актуальность угрозы атаки в том случае, если нарушителем частично при реализации атаки будут использоваться новые неизвестные кому-либо уязвимости.

Таким образом, данный метод моделирования для оценки актуальности угрозы целевой атаки заключается в моделировании невозстанавливаемого отказа безопасности системы в отношении этой угрозы, что учитывается в модели использованием поглощающего состояния. При этом,

поскольку в такой модели отсутствует стационарный режим, моделируется временная характеристика безопасности — среднее время наработки системы до отказа безопасности.

3. Моделирование угроз атак в общем виде

В общем случае с учетом возникновения и устранения уязвимостей в процессе работы системы угроза атаки может принимать различные формы, что можно соответствующим образом учесть при ее моделировании, более корректно описав угрозу целевой атаки. Будем рассматривать далее угрозу атаки в общем виде — с учетом ее трансформации в процессе функционирования системы, при условии, что угроза атаки создается угрозами уязвимостей нескольких типов.

Так, изначально при отсутствии уязвимостей их возникновение создает реальную угрозу эксклюзивной целевой атаки. Далее, когда все возникшие уязвимости одного типа будут устранены, а другого типа еще нет, возникновение новой уязвимости того типа, все уязвимости которого были устранены, уже создает реальную угрозу гибридной атаки, поскольку данная уязвимость нулевого дня сразу может быть использована для осуществления атаки, угроза которой становится реальной. Наличие же в системе одновременно уязвимостей всех типов, создающих угрозу атаки, о которых известно, которые устраняются, создает уже реальную угрозу массовой атаки.

Рассмотрим, как подобное изменение формы, принимаемой угрозой атаки в процессе функционирования системы, может быть учтено при моделировании путем построения модели угрозы атаки в общем виде, опять же в предположении о том, что угроза атаки создается угрозами уязвимостей двух типов, которая приведена на рис. 3. Можно говорить, что это модель угрозы атаки в общем виде, поскольку ею учитываются все формы, принимаемые угрозой атаки в процессе функционирования системы.

Поскольку в данном случае нас будет интересовать случайный процесс изменения формы угрозы целевой атаки, и при этом все уязвимости должны устраняться, естественно, строим модель восстанавливаемой системы, позволяющей определять соответствующие параметры, вероятностные и временные характеристики угрозы атаки для стационарного режима функционирования системы.

Состояние S_{00} в модели — это состояние, характеризующее отсутствие

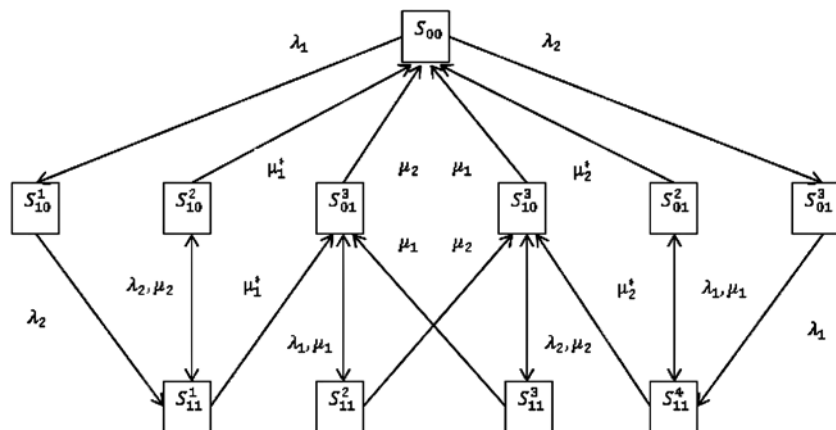


Рис. 3. Иллюстрация модели угрозы атаки в общем виде

уязвимостей в системе, S_{10}^i — состояния, характеризующие присутствие в системе, по крайней мере, одной уязвимости первого типа, S_{01}^i — состояния, характеризующие присутствие в системе, по крайней мере, одной уязвимости второго типа, S_{11}^j — состояния, характеризующие присутствие в системе, по крайней мере, одной уязвимости первого типа и, по крайней мере, одной уязвимости второго типа, угроза атаки в этих состояниях реальна.

Отметим, что данная модель соответствует неразложимой неперриодической цепи Маркова с конечным числом состояний, как следствие, она имеет стационарный режим функционирования и может характеризоваться соответствующими финальными вероятностями состояний.

Рассмотрим данную модель, при этом поясним, почему потребовалось ввести в модель i состояний S_{10} и S_{01} и j состояний S_{11} , и определимся с тем, как и в результате чего меняется форма целевой атаки. Анализ модели будем проводить исходя из того, что система осуществляет переход $S_{00} \rightarrow S_{10}^1$ (в случае перехода $S_{00} \rightarrow S_{01}^3$ анализ проводится полностью аналогично).

Переход $S_{00} \rightarrow S_{10}^1$ предполагает возникновение уязвимости нулевого дня первого типа, при этом данная уязвимость потенциальным нарушителем не используется (ему неизвестна уязвимость второго типа). При нахождении системы в состоянии S_{10}^1 может возникнуть несколько уязвимостей первого типа (имеем угрозу отказа безопасности [5]), что требуется учесть в модели. Переход $S_{10}^1 \rightarrow S_{11}^1$ предполагает возникновение реальной эксклюзивной угрозы целевой атаки, которая будет реализована, в результате чего об уязвимостях нулевого дня каждого типа становится известно, они начинают устраняться. В результате этого угроза атаки становится массовой. В это время могут возникать новые уязвимости обоих типов, появление которых повлияет на продолжение пребывания системы в состоянии S_{11}^1 . При возникновении же новой возникшей уязвимости нулевого дня какого-либо типа при пребывании системы в состоянии S_{11}^1 угроза этой целевой атаки становится гибридной. В зависимости от того, все возникшие уязвимости какого типа будут устранены раньше, система из состояния S_{11}^1 перейдет в состояние S_{10}^2 , либо в состояние S_{01}^3 . И так далее.

Теперь рассмотрим, с какой целью, кроме состояния S_{10}^2 , дополнительно в модель включено состояние S_{10}^3 . Это обусловлено различными интенсивностями (интенсивность восстановления отказа безопасности [5]) возврата системы из этих состояний, соответственно, μ_1 и μ_1^* . Интенсивностью μ_1^* учитывается то, что при нахождении си-

стемы в состоянии S_{10}^1 может возникнуть несколько уязвимостей первого типа нулевого дня, при этом необходимо учесть устранение их всех в системе. В состоянии S_{10}^1 эти уязвимости не устраняются, так как их устранение начинается после осуществления эксклюзивной целевой атаки, когда о них становится известно. Устраняются эти уязвимости при нахождении системы в состояниях S_{10}^2 и S_{11}^1 , что учитывается использованием интенсивности μ_1^* . Переход же $S_{01}^3 \rightarrow S_{11}^2$ предполагает, что все накопленные ранее уязвимости нулевого дня первого типа в состоянии S_{10}^1 и уязвимости этого типа, возникшие при пребывании системы в состоянии S_{11}^1 , устранены, поэтому для перехода $S_{01}^3 \rightarrow S_{00}$ уже учитывается интенсивность μ_1 (интенсивность устранения уязвимости).

Используем для обозначения стационарных вероятностей состояний те же индексы, что и для обозначения соответствующих состояний.

В предположении о том, что накопления уязвимостей в состоянии S_{10}^1 не происходит, исходя из стационарности режима функционирования системы можем записать:

$$P_{10}^1 \lambda_2 + (P_{10}^2 + P_{11}^1) \lambda_1 = (P_{10}^2 + P_{11}^1) \mu_1.$$

Если же мы учтем накопление уязвимостей в состоянии S_{10}^1 , которые должны быть устранены при нахождении системы в состояниях S_{10}^2 и S_{11}^1 , то соответственно получим

$$P_{10}^1 \lambda_2 + (P_{10}^2 + P_{11}^1 + P_{10}^1) \lambda_1 = (P_{10}^2 + P_{11}^1) \mu_1,$$

откуда μ_1^* можем определить следующим образом:

$$\mu_1^* = \mu_1 - \frac{P_{10}^1 \lambda_1}{P_{10}^2 + P_{11}^1}.$$

Аналогично для μ_2^* можем записать:

$$\mu_2^* = \mu_2 - \frac{P_{01}^3 \lambda_2}{P_{01}^2 + P_{11}^4}.$$

Построив подобную модель, уже можно рассчитать все искомые параметры и характеристики угрозы целевой атаки в общем виде с учетом различных принимаемых ею форм в процессе функционирования информационной системы: интенсивности возникновения в системе отказов безопасности (по крайней мере, одной реальной угрозы атаки) λ_a , интенсивность устранения в системе отказов безопасности (всех одновременно присутствующих реальных угроз атак) μ_a , вероятность готовности системы к безопасной эксплуатации P_{0a} , среднее время наработки на отказ безопасности информационной системы (восстанавливаемая система) в отношении угрозы атаки

T_o , среднее время восстановления безопасности информационной системы T_b в отношении угрозы атаки, что позволит корректно количественно оценить уровень ее актуальности. Например, для модели, приведенной на рис. 3, имеем:

$$P_{0a} = 1 - P_{11}^1 - P_{11}^2 - P_{11}^3 - P_{11}^4;$$

$$\lambda_a = (P_{01}^1 + P_{01}^2 + P_{01}^3)\lambda_1 + (P_{10}^1 + P_{10}^2 + P_{10}^3)\lambda_2;$$

$$\mu_a = \frac{(P_{01}^1 + P_{01}^2 + P_{01}^3)\lambda_1 + (P_{10}^1 + P_{10}^2 + P_{10}^3)\lambda_2}{P_{11}^1 + P_{11}^2 + P_{11}^3 + P_{11}^4};$$

$$T_b = \frac{1}{\mu_a}, T_o = \frac{1}{\lambda_a} - T_b.$$

Кроме того, данная модель позволяет получить и ряд иных важных параметров и характеристик, уже связанных с изменением формы угрозы атаки в процессе функционирования системы. Например, интенсивность $\lambda_{\text{цн}}$ возникновения в системе реальных угроз эксклюзивной целевой атаки, предполагающей использование уязвимостей нулевого дня, можно для модели, приведенной на рис. 3, определить следующим образом:

$$\lambda_{\text{цн}} = P_{10}^1\lambda_2 + P_{01}^3\lambda_1,$$

интенсивность же $\lambda_{\text{цн}}$ возникновения в системе реальных угроз гибридной целевой атаки:

$$\lambda_{\text{цн}} = (P_{10}^2 + P_{10}^3)\lambda_2 + (P_{01}^2 + P_{01}^3)\lambda_1.$$

Таким образом, предложенный подход к моделированию позволяет определять все требуемые параметры и характеристики угрозы целевой атаки, позволяющие количественно оценивать уровень ее актуальности, с учетом всех этапов ее трансформации.

Список литературы

1. **Рынок** ищет способы защиты от целевых атак. URL: http://www.cnews.ru/reviews/security2014/articles/rynok_ishchet_sposoby_zashchity_ot_tselevyh_atak, свободный (02.04.2017).
2. **Таргетированные** атаки и как с ними бороться. URL: <https://www.iemag.ru/analytics/detail.php?ID=32831>, свободный (02.04.2017).
3. **Целевые** атаки (APT). URL: <https://www.anti-malware.ru/threats/target-attack>, свободный (02.04.2017).
4. **Целевые** атаки — угроза номер один. Вестник цифровой трансформации CIO.RU. URL: <http://www.cio.ru/articles/724>, свободный (02.04.2017).
5. **Щеглов А. Ю., Щеглов К. А.** Защита информации: основы теории: учебник для бакалавриата и магистратуры. М.: Издательство Юрайт, 2017. 309 с.

К. А. Shcheglov, Postgraduate Student, skd@npp-itb.spb.ru,
А. Ю. Shcheglov, D. Sc., Professor, info@npp-itb.spb.ru,
 ITMO University, St. Petersburg, 197101, Russian Federation

Targeted Attacks Threat Modeling

In this article is dedicated to targeted attacks threat modeling approach, which is marked by use of zero-day vulnerabilities by the potential intruder. This feature radically changes the task of modeling, because of software tools vulnerabilities threat reservation absence in such case (which means a system without reservation must be modeled). Security breakups in this case are not managed because of their unknown status (only the potential intruder knows about this vulnerabilities). In common case taking in mind vulnerabilities appearing and eliminating during system work, attack threat can be present in different forms, which needs to be taken into account while threat modeling (making targeted attack threat description more correct). At the beginning threat is known only by potential intruder, then after it's identification the process of threat eliminating begins. After such process system became restorable, which needs to be taken into account while attack threat modeling (attack can turn it's form). In common case while attack implementation potential intruder can use both unknown and known vulnerabilities which create attack threat meanwhile known vulnerability is not yet patched. Let's review how to calculate these attack threat features inside the mathematical model and which attack threat actuality characteristics (measures) can be defined. We will use Markov model with discrete state and continues time to build attack threat model. The suggested targeted attacks threat modeling approach has such advantages as ability to receive actuality quantitate measures without using of any subjective expert decisions.

Keywords: attack threat, targeted attack, actuality, quantitate measure, mathematical modeling

References

1. **Рынок** ищет способы защиты от целевых атак, available at: http://www.cnews.ru/reviews/security2014/articles/rynok_ishchet_sposoby_zashchity_ot_tselevyh_atak, svobodnyj (date of access: 02.04.2017).
2. **Таргетированные** атаки и как с ними борот'sya, available at: <https://www.iemag.ru/analytics/detail.php?ID=32831>, svobodnyj (02.04.2017).
3. **Cevelye ataki (APT)**, available at: <https://www.anti-malware.ru/threats/target-attack>, svobodnyj (02.04.2017).
4. **Cevelye ataki — ugroza nomer odin. Vestnik tsifrovoi transformatsii** CIO.RU, available at: <http://www.cio.ru/articles/724>, svobodnyj (02.04.2017).
5. **Shcheglov A. Yu., Shcheglov K. A.** Zashchita informacii: osnovy teorii: uchebnyk dlya bakalavriata i magistratury (Information security: base theory: bachelor and magister materials), Moscow, Yurajt, 2017, 309 p. (in Russian).

В. Н. Дам, аспирант, damvan.nhich@gmail.com,
Московский физико-технический институт

OFDM-модуляция в задаче автоматического распознавания вида цифровой модуляции

Исследуется задача включения OFDM-модуляции в набор распознаваемых видов цифровой модуляции радиосигналов с помощью многослойной нейронной сети. Распознавание проводится с использованием кумулянтных признаков при известном значении частоты несущего сигнала. Результаты тестирования показывают, что включение OFDM-модуляции в число распознаваемых видов цифровой модуляции не влияет на вероятности распознавания других видов модуляции, при этом вероятность распознавания OFDM-модуляции практически равна 100 %.

Ключевые слова: OFDM-модуляция, распознавание вида цифровой модуляции, моменты, кумулянты, гауссовый шум, многослойная нейронная сеть, характеристическая функция, IQ данные

Введение

Модуляция является обязательным и важным компонентом коммуникационной системы. Задача распознавания видов цифровой модуляции используется не только в гражданской, но и в военной области, например, в радиоразведке. Распознавание вида цифровой модуляции в зависимости от задач позволяет демодулировать сигнал, выполнить частотную синхронизацию и т. д. В работах [1–4] была исследована задача распознавания следующих видов одночастотной модуляции: M-PSK, M-FSK, M-QAM.

В работе [1] приведены результаты по распознаванию различных видов одночастотной модуляции (M-PSK, M-FSK, M-QAM), при этом вероятность распознавания не превышает 0,8 и вид OFDM-модуляции не включен в число распознаваемых. В работе [2] авторы тоже исследовали виды одночастотной модуляции и доказали, что нейросетевой подход с использованием кумулянтных признаков является наилучшим подходом в задаче распознавания видов цифровой модуляции.

В работе [6] авторы исследовали метод и разработали устройство распознавания видов модуляции. Этот метод позволяет распознавать только два класса сигналов: с одночастотной модуляцией и многочастотной модуляцией (OFDM). Сейчас OFDM (Orthogonal Frequency Division

Multiplexing) модуляцию широко применяют в проводной (ADSL, DVB-C2, PLC HomePlug AV, ...) и беспроводной связи (стандарты IEEE 802.11, HIPERLAN/2, LTE, WiMAX, ...). Основным преимуществом OFDM-модуляции по сравнению с видами одночастотной модуляции является ее способность противостоять сложным условиям в канале. Кроме того, низкая символьная скорость делает возможным использование защитного интервала между символами, что позволяет справляться с временным рассеянием и устранять межсимвольную интерференцию.

В настоящей работе исследуется актуальная задача включения OFDM-модуляции в набор распознаваемых видов цифровой модуляции радиосигналов, который был исследован в работе [5].

1. Постановка задачи

Отправитель передает сигнал с одним из следующих видов цифровой модуляции: 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM и OFDM. Получатель принимает искаженный аддитивными комплексными гауссовыми белыми шумами $\xi(t)$ переданный сигнал $z(t)$, $t = \overline{0, T}$,

$$z(t) = a(t) + \xi(t),$$

где $a(t)$ — переданный сигнал без шума.

Принимаемый сигнал проходит через цифровой понижающий преобразователь (*digital down converter*), и на выходе этого преобразователя получаем сигналы I (синфазную составляющую) и Q (квадратурную составляющую), которые являются исходными данными для поставленной задачи. Получателю известна частота дискретизации сигнала, несущая частота и параметры OFDM-модуляции, которые представлены в разд. 4, табл. 4.

Получателю требуется распознавать используемый вид цифровой модуляции.

2. Основы OFDM-модуляции

OFDM-модуляция (Orthogonal Frequency Division Multiplexing) — цифровая схема модуляции, которая использует большое число близко расположенных ортогональных поднесущих. OFDM-модуляция является специальным случаем передачи сигналов со многими несущими, когда данные из последовательного потока передаются одновременно (параллельно) через совокупность поднесущих. Каждая поднесущая модулируется по обычной схеме модуляции (фазовой или квадратурной амплитудной).

Для формирования OFDM-модуляции используется обратное быстрое преобразование Фурье (ОБПФ), а демодуляции — прямое быстрое преобразование Фурье (БПФ). На рис. 1, 2 представлены структурные схемы формирования OFDM-сигнала передатчика и приемника [7].

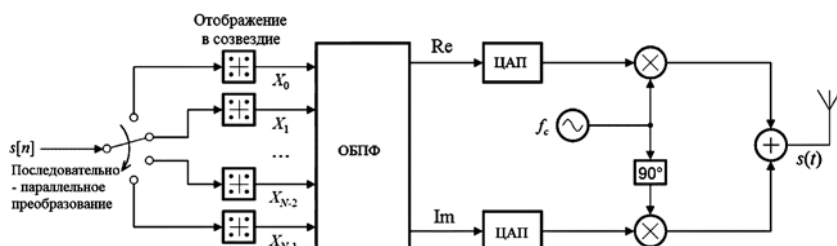


Рис. 1. Формирование OFDM-сигнала передатчика

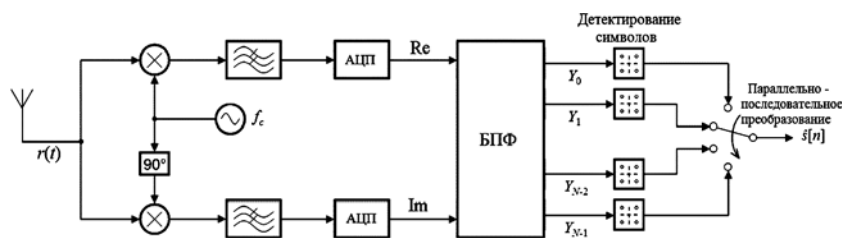


Рис. 2. Формирование OFDM-сигнала приемника



Рис. 3. Система автоматического распознавания вида цифровой модуляции

OFDM-символ состоит из отчетов сигнала после ОБПФ. Для исключения межсимвольной интерференции в каждый OFDM-символ вводится защитный интервал, длительность которого может составлять 1/4, 1/8, 1/16, 1/32 от длительности OFDM-символа: $T_s = T_d + T_g$, где T_s — длительность OFDM-символа; T_d — длительность символа после ОБПФ; T_g — длительность защитного интервала.

3. Система распознавания вида цифровой модуляции

В системе распознавания вида цифровой модуляции ключевым звеном является обученная многослойная нейронная сеть (МНС). Обучение МНС выполняется подачей на ее вход признаков, которые характеризуют конкретный вид модуляции, указанием вида модуляции, например, ее номера в списке распознаваемых видов модуляции, и последующей коррекцией памяти МНС. Выбор информативных признаков определяет успех или неуспех решения задачи распознавания. Предыдущие исследования [5] показали, что наиболее эффективным подходом к выбору информативных признаков служит кумулянтный подход, который и в настоящей работе применяется для распознавания OFDM-модуляции.

Модель системы распознавания вида цифровой модуляции представлена на рис. 3.

Система состоит из трех частей: выделение синфазной (I) и квадратурной (Q) составляющих сигнала; вычисление информативных признаков; нейронная сеть, обученная распознаванию всех видов цифровой модуляции, приведенных в работе [5], а также и OFDM-модуляции.

3.1. Выделение синфазной и квадратурной составляющих сигнала

На рис. 4 приведена схема обработки получаемого сигнала. Выходами блока являются IQ -сигналы, которые поступают в следующий блок для вычисления информативных признаков.

3.2. Информативные признаки

При конструировании кумулянтных признаков как цифровые модулированные сигналы, так и шумы в системе связи далее рассматриваются как стационарные случайные процессы. Характеристиками случайных процессов, в частности, служат мо-

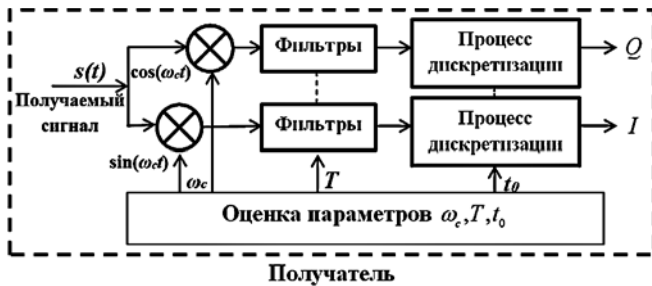


Рис. 4. Схема обработки получаемого сигнала

менты различных порядков и связанные с ними кумулянты (семиинварианты). В качестве информативных признаков далее используются кумулянтные признаки, определение которых основано на вычислении моментов случайного процесса. Поэтому далее приводится краткая информация о методах вычисления кумулянтов и их свойствах.

При использовании кумулянтных признаков получаемые сигналы преобразуются в низкочастотные сигналы (IQ -сигналы), что удобно для обработки цифровых сигналов.

Вычисление кумулянтов одномерного процесса. Пусть имеется случайная величина ξ с плотностью распределения $f_\xi(x)$.

Моментом первого порядка служит величина $a = \int_{-\infty}^{\infty} x f_\xi(x) dx$. Центрированные моменты порядка n вычисляют по формуле

$$E_n = \int_{-\infty}^{\infty} (x - a)^n f_\xi(x) dx.$$

Если момент первого порядка равен нулю, то моменты n -го порядка равны

$$E_n = \int_{-\infty}^{\infty} x^n f_\xi(x) dx.$$

Вычисление кумулянтов связано с характеристической функцией случайной величины ξ , которая определяется как математическое ожидание случайной величины $\eta = e^{iu\xi}$ по формуле

$$\varphi(u) = \int_{-\infty}^{\infty} e^{iux} f_\xi(x) dx.$$

Характеристическая функция обладает целым рядом полезных и важных свойств, что и объясняет ту роль, которую она играет в теории математической статистики. Кумулянтами случайной величины служат коэффициенты C_n разложения логарифма характеристической функции $\varphi(u)$ в ряд Тейлора:

$$\ln \varphi(u) = \sum_{n=0}^{\infty} C_n (iu)^n / n!,$$

где n — порядок кумулянта.

Известно, что между кумулянтами и моментами случайной величины существуют прямая и обратная зависимость [8]. Кумулянты вычисляются с помощью n -кратного дифференцирования характеристической функции. Ниже приведены формулы для вычисления первых восьми кумулянтов для одномерной центрированной случайной величины через соответствующие моменты [8]:

$$\begin{aligned} C_2 &= E_2; \\ C_3 &= E_3; \\ C_4 &= E_4 - 3E_2^2; \\ C_5 &= E_5 - 10E_3E_2; \\ C_6 &= E_6 - 15E_4E_2 - 10E_3^2 + 30E_2^3; \\ C_7 &= E_7 - 21E_5E_2 - 35E_4E_3 + 210E_3E_2^2; \\ C_8 &= E_8 - 28E_6E_2 - 56E_5E_3 - 35E_4^2 + \\ &+ 420E_4E_2^2 + 560E_3^2E_2 - 630E_2^4. \end{aligned} \quad (1)$$

Вычисление кумулянтов для двумерного процесса. Согласно рис. 2 в результате обратного преобразования Фурье получаемый OFDM-сигнал является комплексным, т. е. двухкомпонентным, и поэтому и моменты, и кумулянты должны быть вычислены для двумерной случайной величины.

Рассмотрим случайный центрированный комплексный модулированный сигнал $s(t) = I(t) + jQ(t)$ дискретного времени $t = 1, 2, \dots$ и комплексно-сопряженный к нему сигнал $\bar{s}(t) = I(t) - jQ(t)$. Таким сигналам соответствует двумерная плотность распределения вероятностей $f(x, y)$ и соответствующая двумерная характеристическая функция $\varphi(u, v) = \iint e^{j(ux+vy)} f(x, y) dx dy$.

Двумерная плотность распределения вероятностей $f(x, y)$ позволяет вычислить центрированные смешанные моменты $E_{k,l} = \iint x^k y^l f(x, y) dx dy$ и их оценки по формуле

$$\hat{E}_{k,l} = \sum_{t=1}^N s^k[t] \bar{s}^l[t] / N, \quad (2)$$

где N — число отсчетов процессов $s(t)$ и $\bar{s}(t)$. Значения смешанных моментов будут в дальнейшем использованы для вычисления соответствующих смешанных кумулянтов.

Смешанными кумулянтами процессов $s(t)$ и $\bar{s}(t)$ служат коэффициенты $C_{n,m}$ разложения логарифма характеристической функции $\varphi(u, v)$ в ряд Тейлора:

$$\ln \varphi(u, v) = \sum_{n,m=0}^{\infty} C_{n,m} (iu)^n (iv)^m / n! m!. \quad (3)$$

Порядком смешанного кумулянта $C_{n,m}$ является сумма $r = n + m$.

Формулы вычисления смешанных кумулянтов выводятся из формул кумулянтов одномерного процесса посредством формальной процедуры, приведенной в работе [8]. Суть ее покажем на примере вычисления смешанного кумулянта $C_{3,1}$, используя формулу для кумулянта $C_4 = E_4 - 3E_{2,0}^2$ из формул (1). Нетрудно показать, что для двухмерного случая данная формула имеет вид $C_{4,0} = E_{4,0} - 3E_{2,0}^2$, которую в формальном виде для первого процесса $s(t)$ запишем так:

$$C(r^4) = E(r^4) - 3[E(r^2)]^2, \quad (4)$$

где переменная r соответствует первому процессу $s(t)$. Далее к обеим частям формулы (4) применяем оператор дифференцирования $q \frac{d}{dr}$, где переменная q соответствует второму процессу $\bar{s}(t)$:

$$4C(r^3s) = 4E(r^3s) - 12E(r^2)E(rs);$$

$$C(r^3s) = E(r^3s) - 3E(r^2)E(rs).$$

Из второго соотношения следует формула для вычисления смешанного кумулянта $C_{3,1}$:

$$C_{3,1} = E_{3,1} - 3E_{2,0}E_{1,1},$$

в которой индексы соответствующих переменных совпадают со степенями переменных r и q .

Применяя этот формальный подход к другим кумулянтам одномерного процесса, можно получить формулы вычисления других смешанных кумулянтов. В табл. 1 представлены формулы для вычисления только тех кумулянтов до восьмого порядка, которые необходимы для решения задачи распознавания вида цифровой модуляции.

Формулы для вычисления кумулянтов

Кумулянт	Выражение через моменты
$C_{2,0}$	$C_{2,0} = E_{2,0}$
$C_{4,0}$	$E_{4,0} - 3E_{2,0}^2$
$C_{3,1}$	$E_{3,1} - 3E_{2,0}E_{1,1}$
$C_{2,2}$	$E_{2,2} - E_{2,0}E_{0,2} - 2E_{1,1}^2$
$C_{6,0}$	$E_{6,0} - 15E_{4,0}E_{2,0} - 10E_{3,0}^2 + 30E_{2,0}^3$
$C_{5,1}$	$E_{5,1} - 5E_{4,0}E_{1,1} - 10E_{3,1}E_{2,0} - 10E_{3,0}E_{2,1} + 30E_{2,0}^2E_{1,1}$
$C_{4,2}$	$E_{4,2} - E_{4,0}E_{0,2} - 8E_{3,1}E_{1,1} - 4E_{3,0}E_{1,2} - 6E_{2,2}E_{2,0} - 6E_{2,1}^2 + 6E_{2,0}^2E_{0,2} + 24E_{2,0}E_{1,1}^2$
$C_{3,3}$	$E_{3,3} - 3E_{3,1}E_{0,2} - E_{3,0}E_{0,3} - 9E_{2,2}E_{1,1} - 9E_{2,1}E_{1,2} - 3E_{2,0}E_{1,3} + 18E_{2,0}E_{1,1}E_{0,2} + 12E_{1,1}^3$
$C_{8,0}$	$E_{8,0} - 28E_{6,0}E_{2,0} - 56E_{5,0}E_{3,0} - 35E_{4,0}^2 + 420E_{4,0}E_{2,0}^2 + 560E_{3,0}^2E_{2,0} - 630E_{2,0}^4$

Значения кумулянтов являются комплексными числами. Основные отличия значений кумулянтов для различных видов цифровой модуляции проявляются в значениях их действительных частей, поэтому в качестве распознающего признака здесь приняты значения действительных частей кумулянтов.

Кумулянты чистого сигнала, шума и получаемого сигнала. Основной причиной использования в качестве информативных признаков в настоящей работе кумулянтных признаков является то, что кумулянты выше второго порядка для гауссового процесса равны нулю [9], а шумы в сетях связи в большинстве случаев носят гауссовый характер.

Рассмотрим структуру характеристической функции для случая, когда наблюдаемые комплексные процессы $s(t)$ и $\bar{s}(t)$ представляют собой сумму чистого сигнала $a(t)$ и распределенного по нормальному закону шума $\eta(t)$: $s(t) = a(t) + \eta(t)$. Характеристическая функция двумерной случайной величины $s(t)$ и $\bar{s}(t)$ равна

$$\varphi_{s, \bar{s}}(u, v) = M(e^{jus + jv\bar{s}}) = M(e^{ju(a+\eta) + jv(\bar{a}+\bar{\eta})}) = M(e^{j(ua+v\bar{a}) + j(u\eta+v\bar{\eta})}) = M(e^{j(ua+v\bar{a})} e^{j(u\eta+v\bar{\eta})}). \quad (5)$$

Если случайные процессы независимы, то математическое ожидание их произведения равно произведению их математических ожиданий, и, следовательно, характеристическая функция (5) равна произведению характеристических функций чистого сигнала и шума:

$$\varphi_{s, \bar{s}}(u, v) = \varphi_{a, \bar{a}}(u, v) \varphi_{\eta, \bar{\eta}}(u, v). \quad (6)$$

Поскольку кумулянтами являются коэффициенты разложения логарифма характеристической функции в ряд Тейлора, то из (6) следует, что кумулянт наблюдаемого процесса равен сумме кумулянта чистого сигнала $a(t)$ и кумулянта шума $\eta(t)$ в силу следующего выражения:

$$\ln \varphi_{s, \bar{s}}(u, v) = \ln \varphi_{a, \bar{a}}(u, v) + \ln \varphi_{\eta, \bar{\eta}}(u, v). \quad (7)$$

Соотношение (7) подтверждает тезис начала данного раздела о том, что в случае нормального шума значения кумулянтов наблюдаемого сигнала выше второго порядка не зависят от уровня шума, искажающего полезный сигнал.

Используя предложенные в работе [5] правила выбора кумулянтных признаков, в настоящей работе в качестве информативных признаков для решения задачи распознавания вида цифровой модуляции приняты девять смешанных кумулянтов: $C_{2,0}$, $C_{4,0}$, $C_{3,1}$, $C_{2,2}$, $C_{6,0}$, $C_{5,1}$, $C_{4,2}$, $C_{3,3}$, $C_{8,0}$.

Правилами, по которым выбраны кумулянты в качестве информативных признаков, являются следующие: а) действительные значения кумулянтов должны быть значимыми; б) значения кумулянтов должны быть различными для разных видов цифровой модуляции; в) значения кумулянтов не должны зависеть от уровня шума, т.е. значения кумулянтов не должны значительно изменяться, когда изменяются значения отношения сигнала к шуму.

3.3. Обучение нейронной сети

Программирование процессов распознавания видов цифровой модуляции выполнено в системе MATLAB с использованием пакетов прикладных программ Communications System Toolbox, Filter Design and Analysis Toolbox и Neural Network Toolbox, входящих в состав MATLAB Application Toolboxes. Для составления базы данных для обучения и тестирования нейронной сети были сгенерированы 8000 сигналов длительностью 3600 отсчетов (1000 сигналов по каждому виду модуляции). Из них 7200 сигналов для обучения и 800 — для тестирования. Входом в нейронную сеть служит девятимерный вектор, компоненты которого составлены из оценок кумулянтов, приведенных в табл. 1. Оценки кумулянтов вычисляются через оценки соответствующих моментов по формуле (2) для каждого входного сигнала. Таким образом, входной процесс обучения представлен матрицей размером 9×7200 , а выходной процесс — матрицей размером 8×7200 для восьми распознаваемых видов цифровой модуляции.

Процесс обучения проводится в режиме offline, максимальное число эпох равно 1000, алгоритм обучения — Levenberg — Marquardt. Ниже приведены параметры нейронной сети:

Число слоев	3
Число входов и нейронов в первом слое	9
Число нейронов в скрытом слое	80
Число нейронов в выходном слое	8
Максимальное число эпох	1000
Функция активации в скрытом слое . . .	сигмоидальная
Функция активации в выходном слое . .	сигмоидальная

4. Результаты тестирования при расширенном наборе распознаваемых видов цифровой модуляции

В табл. 2 представлены результаты распознавания без включения OFDM-модуляции [5].

Таблица 2

Результаты (в процентах) распознавания вида цифровой модуляции

Вид модуляции	2PSK	4PSK	8PSK	2FSK	8QAM	16QAM	64QAM
Вероятность распознавания, %	100	100	100	100	100	88,0	87,5

Таблица 3

Параметры OFDM-сигналов

Вариант	1	2	3	4
Число поднесущих	101	101	225	225
Длина ОБПФ	128	128	256	256
Схема модуляции в поднесущих	64-QAM	16-QAM	16-QAM	64-QAM
Частота дискретизации, Гц	839 680	839 680	839 680	839 680
Защитный интервал	16	16	32	32

Таблица 4

Результаты распознавания видов цифровой модуляции при различных вариантах OFDM-сигналов

Вид модуляции	Вероятность распознавания, %			
	1-й вариант	2-й вариант	3-й вариант	4-й вариант
2-PSK	100	100	100	100
4-PSK	100	100	100	100
8-PSK	100	100	100	100
2-FSK	100	100	100	100
8-QAM	100	100	100	100
16-QAM	88,1	86,2	80,5	84,5
64-QAM	90,7	90,8	88,1	88,2
OFDM	100	100	100	100

В табл. 3 приведены различные параметры OFDM-сигналов, которые используются для тестирования. Все тестируемые сигналы имеют отношение сигнала к нормальному шуму 20 дБ.

В табл. 4 представлены результаты тестирования при различных вариантах OFDM-сигналов.

Результаты тестирования, приведенные в табл. 2, 4, показывают, что включение вида OFDM в набор видов модуляции не влияет на вероятности распознавания других видов. При этом вероятность распознавания OFDM-модуляции практически равна 100 %.

Заключение

В данной работе приведены результаты расширения набора распознаваемых видов цифровой модуляции при включении OFDM-модуляции

в набор распознаваемых видов цифровой модуляции. Приведенные результаты показывают эффективность использования кумулянтов принимаемых сигналов в качестве признаков для обучения, тестирования и последующего функционирования многослойной нейронной сети в условиях гауссова шума. Следует отметить, что расширение набора распознаваемых видов цифровой модуляции при включении OFDM-модуляции практически не привело к уменьшению вероятности распознавания других видов цифровой модуляции, распознавание которых в большинстве случаев выполняется с высокой вероятностью, близкой к 100 %.

Все приведенные результаты получены при известном значении несущей частоты. В настоящее время проводятся исследования, направленные на решение задачи при неизвестном значении несущей частоты и применение полученных результатов в конкретной реальной системе коммуникации.

Список литературы

1. **Аджемов С. С., Терешонок М. В.** Способ и устройство автоматического распознавания типов манипуляции радиосигналов. Патент № RU2510077C2. 2014 г.
2. **Аджемов С. С., Кленов Н. В., Терешонок М. В., Чиров Д. С.** Методы распознавания видов цифровой модуляции сигналов в когнитивных радиосистемах // Вестник Московского Университета. Серия 3. Физика. Астрономия. 2015. № 6. С. 19—27.
3. **El-Khoribi R. A., Shoman M. A. I., Mohammed A. G. A.** Automatic Digital Modulation Recognition using Artificial Neural Network in Cognitive radio // International Journal of Emerging Trends & Technology in Computer Science (IJETTCS). May — June, 2014. Vol. 3. Is. 3. P. 132—136.
4. **Liu M., Zhao Y., Shi L., Dong J.** Research on recognition algorithm of digital modulation by higher order cumulants // Proc. of 4-th International Conference on Instrumentation and Measurement, Computer, Communication and Control. 2014. P. 686—690.
5. **Дам В. Н.** Автоматическое распознавание видов цифровой модуляции радиосигналов с помощью многослойной нейронной сети по кумулянтным признакам // Информационные технологии. 2016. № 7. Т. 22. С. 555—560.
6. **Peng Liu, Li Zou.** Apparatus and method for classifying modulation in multipath environments. Patent No: US8385473B2. 2010.
7. **Бакулин М. Г., Крейнделин В. В., Шлома А. М., Шумов А. П.** Технология OFDM: учеб. пособие для вузов. М.: Горячая линия — Телеком, 2017. 352 с.
8. **Кендалл М. Дж., Стьюарт А.** Теория распределений. Т. 1. М.: Наука, 1966. 587 с.
9. **Гнеденко Б. В.** Курс теории вероятностей. Изд. 6-е. М.: Наука, 1988. 447 с.

Dam Van Nhich, Graduate Student, damvan.nhich@gmail.com,
Department IIST FREC of Moscow Institute of Physics & Technology

The OFDM Modulation in the Problem of Automatic Modulation Recognition

In this paper is explored the problem of including OFDM modulation into the set of digital modulation types of radio signals using the multilayer neural network. Recognition is conducted using the cumulant features with known carrier frequency. The test results show that the inclusion OFDM modulation into the set of digital modulation types does not affect on the recognition probabilities of other types, while the recognizing probability of OFDM type is almost 100 %.

Keywords: OFDM modulation, digital modulation recognition, moments, cumulants, Gaussian noise, multilayer neural network, characteristic function, IQ data

References

1. **Adzhemow C. C., Tereschonok M. W.** *Sposob i ustrojstvo avtomatitscheskogo raspoznaniya tipov manipuljacii radiosignalov*, Patent No. RU2510077C2. 2014 (in Russian).
2. **Adzhemow C. C., Klenow N. W., Tereschonok M. W., Tschirrow D. C.** *Metody raspoznaniya vidov cifrovoj moduljacii signalov w kognitiwnych radiosistemach*, *Vestnik Moskovskogo Universiteta. Serija 3. Fisika. Astronomija*. 2015, no. 6, pp. 19—27 (in Russian).
3. **El-Khoribi R. A., Shoman M. A. I., Mohammed A. G. A.** *Automatic Digital Modulation Recognition using Artificial Neural Network in Cognitive radio*, *Proc. of International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*. May — June, 2014, vol. 3, is. 3, pp. 132—136.
4. **Liu M., Zhao Y., Shi L., Dong J.** *Research on recognition algorithm of digital modulation by higher order cumulants*, *Proc. of 4-th International Conference on Instrumentation and Measurement, Computer, Communication and Control*, 2014, pp. 686—690.
5. **Dam V. N.** *Avtomaticheskoe raspoznaniye vidov cifrovoj moduljacii radiosignalov s pomosh'ju mnogoslojnoj nejronnoj seti po kumuljantnym priznakam*, *Informacionnye tehnologii*, 2016, vol. 22, no. 7, pp. 555—560 (in Russian).
6. **Peng Liu, Li Zou.** *Apparatus and method for classifying modulation in multipath environments*, Patent No. US8385473B2. 2010.
7. **Bakulin M. G., Krejndelin V. V., Shloma A. M., Shumov A. P.** *Tehnologija OFDM, Uchebnoe posobie dlja vusov*, Moscow, Goryachaja linya — Telekom, 2017, 352 p. (in Russian).
8. **Kendall M. Dzh., St'juart A.** *Teoriya raspredelenij*, vol. 1, Moscow, Nauka, 1966, 587 p. (in Russian).
9. **Gnedenko B. W.** *Kurs teorii veroyatnostej*. Izd. 6-e, Moscow, Nauka, 1988, 447 p. (in Russian).

А. А. Коляда, д-р физ.-мат. наук, доц., гл. науч. сотр., **П. В. Кучинский**, д-р физ.-мат. наук, проф.,
Научно-исследовательское учреждение "Институт прикладных физических проблем
имени А. Н. Севченко" Белорусского государственного университета, Минск, Беларусь,
Н. И. Червяков, д-р техн. наук, проф., e-mail: razan@tut.by, niipfp@bsu.by, Chervyakov@yandex.ru,
Северо-Кавказский федеральный университет, Ставрополь, РФ

Редукционный метод позиционно-модулярного преобразования больших чисел для нейронных сетей на конечных кольцах

Рассматривается проблема построения нейронных сетей конечного кольца (НСКК), которые служат основой нейросетевых модулярных вычислительных структур для высокопроизводительных криптографических приложений. Методологическую базу НСКК исследуемого класса составляет модифицированный редукционный метод позиционно-модулярного преобразования взвешенных больших чисел. Дана математическая формализация метода, получены оценки диапазона изменения и разрядности элементов последовательности вычетов, формируемой по применяемой редукционной схеме рекурсивного типа, исследованы характер и скорость ее сходимости, предложен гибкий табличный механизм сокращения числа итераций схемы. Синтезирован общий редукционный алгоритм позиционно-модулярного кодового преобразования, разработана параллельная структура НСКК, осуществляющей базовое преобразование за одну итерацию.

Ключевые слова: нейронная сеть, нейронная сеть конечного кольца, синаптические веса, модулярная система счисления, модулярная арифметика, криптография, диапазон больших чисел, редукционный метод позиционно-модулярного преобразования

Введение

В современном процессе развития эффективных средств защиты информации фундаментальная роль отводится разработкам по созданию новых вычислительных технологий, ориентированных на высокоскоростную реализацию трудоемких базовых процедур в диапазонах больших чисел (ДБЧ) [1–8]. С точки зрения производительности при оперировании на ДБЧ приоритетные позиции принадлежат модулярным вычислительным технологиям. Важнейшим фактором, способствующим неуклонному повышению уровня востребованности данных технологий, является их идеальная приспособленность к нейросетевым реализациям [1, 3, 9–13]. Активно развиваемое в настоящее время новое направление в криптографии — разработка и оптимизация нейросетевых модулярных вычислительных структур (МВС) [1, 3], нацеленное на реализацию в максимальной мере оптимально согласованных свойств параллелизма искусственных нейронных сетей (ИНС) и модулярной арифметики (МА),

дает принципиально новые возможности для построения высокопроизводительных криптосистем различного функционального назначения. Отмеченное обстоятельство обусловлено тем, что при согласованном числе синапсов нейронной сети (НС), используемых в процессе взаимодействия ее нейронов, и мощностью базиса применяемой модулярной системы счисления (МСС) НС становится естественным представлением данной числовой системы [12]. На адекватность системы счисления в остатках и НС указывают, в частности, следующие признаки:

- семантическое сходство позиционных форм модулярных чисел с расчетными соотношениями формального нейрона;
- существование адекватного отображения алгоритмов арифметических операций в МСС на многослойные НС;
- простота реализации основных операций нейросетевого логического базиса в модулярном коде;
- равнозначность модулярного кодирования информации и ассоциативной нейронной па-

мости, вытекающая из смыслового (семантического) соответствия оснований МСС классификационному признаку сети, а остаткам по основаниям — значению этого признака.

Основополагающая идея теоретических и прикладных разработок по созданию методологических, алгоритмических и программно-аппаратных средств реализации позиционных форм модулярных чисел [1, 2, 14, 15] состоит в переводе вычислений из ДБЧ в компьютерные диапазоны целых чисел (ЦЧ) стандартной разрядности. Ключевую роль в процессе решения сформулированной задачи выполняют НС, определенные на конечных кольцах вычетов по рабочему базису модулей. Операционную основу нейронных сетей конечного кольца (НСКК) составляют главным образом операции приведения целых чисел к остаткам по используемым модулям. Как структурно, так и на операционном уровне НСКК в максимальной мере должны быть согласованы с естественным кодовым параллелизмом МА. В полной мере данному условию удовлетворяет рассматриваемый в настоящей работе редуцированный метод позиционно-модулярного кодового преобразования.

1. Математическая формализация редуцированного метода позиционно-модулярного преобразования больших чисел

Введем обозначения:

- \mathbf{Z} — множество целых чисел;
- $\lfloor a \rfloor$ и $\lceil a \rceil$ — наибольшее и наименьшее ЦЧ соответственно, не большее и не меньшее вещественной величины a ;
- $\mathbf{Z}_m = \{0, 1, \dots, m-1\}$, $\mathbf{Z}_m^- = \{-\lfloor m/2 \rfloor, -\lfloor m/2 \rfloor + 1, \dots, \lfloor m/2 \rfloor - 1\}$ — множества наименьших неотрицательных и абсолютно наименьших вычетов по натуральному модулю m ;
- $|a|_m$ и $|a|_m^-$ — элементы множеств \mathbf{Z}_m и \mathbf{Z}_m^- , сравнимые с a (в общем случае рациональным числом) по модулю m .

Функциональное назначение НС конечного кольца по модулю m исследуемого класса состоит в вычислении остатка от деления произведения CX на m ,

$$\chi = |CX|_m, \quad (1)$$

где C — целочисленная константа; X — входное неотрицательное ЦЧ, представленное b -разрядным двоичным кодом $(x_{b-1} x_{b-2} \dots x_0)_2$ ($x_j \in \{0, 1\}$, $j = \overline{0, b-1}$). По критерию простоты нейросетевой реализации наиболее приемлемым методом вы-

полнения операции (1) является метод модулярной редукции суммы взвешенных операндов по рекурсивной схеме последовательного снижения разрядности получаемых вычетов [1, 3].

Положим

$$\begin{aligned} X^{(0)} &= (x_{b_0-1}^{(0)} x_{b_0-2}^{(0)} \dots x_0^{(0)})_2 = \\ &= \sum_{j=0}^{b_0-1} 2^j x_j^{(0)} \quad (b_0 = b, x_j^{(0)} = x_j) \end{aligned} \quad (2)$$

и пусть

$$\begin{aligned} W_j(C) &= \left| C \cdot 2^j \right|_{m_0}^- = \\ &= \begin{cases} \left| C \cdot 2^j \right|_m, & \text{если } \left| C \cdot 2^j \right|_m < \left\lceil \frac{m}{2} \right\rceil, \\ \left| C \cdot 2^j \right|_m - m, & \text{если } \left| C \cdot 2^j \right|_m \geq \left\lceil \frac{m}{2} \right\rceil, \end{cases} \quad j = \overline{0, b-1}. \end{aligned} \quad (3)$$

При $C = 1$ далее используется обозначение $W_j = W_j(1)$.

Применяемая редуцированная схема описывается операционной последовательностью:

$$\begin{aligned} \left\langle X^{(1)} &= \sum_{j=0}^{b_0-1} W_j(C) x_j^{(0)} = \right. \\ &= (x_{b_1-1}^{(1)} x_{b_1-2}^{(1)} \dots x_0^{(1)})_2 - 2^{b_1} x_{b_1-1}^{(1)} = \\ &= \sum_{j=0}^{b_1-2} 2^j x_j^{(1)} - 2^{b_1-1} x_{b_1-1}^{(1)}, \\ X^{(s)} &= \sum_{j=0}^{b_{s-1}-2} W_j x_j^{(s-1)} - W_{b_{s-1}-1} x_{b_{s-1}-1}^{(s-1)} = \\ &= (x_{b_s-1}^{(s)} x_{b_s-2}^{(s)} \dots x_0^{(s)})_2 - 2^{b_s} x_{b_s-1}^{(s)} = \\ &= \sum_{j=0}^{b_s-2} 2^j x_j^{(s)} - 2^{b_s-1} x_{b_s-1}^{(s)} \quad (s = \overline{2, S}); \left. \right\rangle, \\ \chi &= \left| X^{(S)} \right|_m, \end{aligned} \quad (4)$$

где b_1 и b_s — длины дополнительных двоичных кодов $(x_{b_1-1}^{(1)} x_{b_1-2}^{(1)} \dots x_0^{(1)})_2$ и $(x_{b_s-1}^{(s)} x_{b_s-2}^{(s)} \dots x_0^{(s)})_2$ соответственно ЦЧ $X^{(1)}$ и $X^{(s)}$, которые, как следует из (4), в принципе могут быть и отрицательными; S — число итераций схемы.

В целях получения необходимой информационной базы для исследования характера скорости сходимости и оптимизации числа S итераций рекурсивного процесса (4) — преобразований типа $X \rightarrow |CX|_m$, оценим мощность диапазона изменения ЦЧ $X^{(s)}$ и его разрядность b_s .

Предположим, что m — простое число и обозначим его разрядность через $b_{\text{mod}} = \lceil \log_2 m \rceil$ бит.

Как известно [16, 17], множество всех степеней числа 2 по модулю m в мультипликативной группе кольца \mathbf{Z}_m образуют так называемую циклическую подгруппу, порождаемую элементом 2. Порядок N этой подгруппы (число ее элементов) служит делителем функции Эйлера $\varphi(m) = m - 1$. Сказанное относится и к совокупности абсолютных наименьших остатков от деления указанных степеней на m , т. е. к определяемому по формуле (3) набору вычетов:

$$\{W_j \in \mathbf{Z}_m^- | W_j = \overline{2^j} |_{m^-}; j = \overline{0, N-1}\}. \quad (5)$$

При $b_{\text{mod}} < b_{s-1}$ последовательности весовых коэффициентов

$$\{W_0, W_1, \dots, W_{N-1}, W_N, \dots, W_{b_{s-1}-1}\} (s = \overline{2, S}), (6)$$

используемые в (4), имеют циклическую структуру. Сегменты длины N в (6) с начальными элементами W_{iN} ($i = \overline{0, \overline{b_{s-1}/N-1}}$) совпадают с последовательностью (5). В случае, когда b_{s-1} не делится нацело на N , последний $[b_{s-1}/N]$ -й сегмент в (6) оказывается неполным. Пусть N_+ и N_- — количества соответственно положительных и отрицательных вычетов в множестве (5). Тогда с учетом вышеизложенного при $s = \overline{2, S}$ максимально возможное значение числа $X^{(s)}$ (см. (4)) сверху можно оценить следующим образом:

$$\begin{aligned} \max\{X^{(s)}\} &< \frac{b_{s-1}}{N} \sum_{j=0}^{N_+-1} \left(\frac{m-1}{2} - 1\right) = \\ &= \frac{b_{s-1}}{N} \left(\frac{m-1}{2} + \frac{m-1}{2} - N_+ + 1\right) \frac{N_+}{2} = \\ &= \frac{b_{s-1}}{N} (m - N_+) \frac{N_+}{2}. \end{aligned}$$

Аналогично для минимального значения ЦЧ $X^{(s)}$ верна следующая оценка:

$$\begin{aligned} \min\{X^{(s)}\} &> \frac{b_{s-1}}{N} \sum_{j=0}^{N_--1} \left(-\frac{m-1}{2} + j\right) = \\ &= \frac{b_{s-1}}{N} \left(-\frac{m-1}{2} - \frac{m-1}{2} + N_- - 1\right) \frac{N_-}{2} = \\ &= -\frac{b_{s-1}}{N} (m - N_-) \frac{N_-}{2}. \end{aligned}$$

Следовательно,

$$\begin{aligned} \max\{X^{(s)}\} - \min\{X^{(s)}\} &< \\ &< \frac{b_{s-1}}{N} \left(\frac{N_+}{2} (m - N_+) + \frac{N_-}{2} (m - N_-)\right) = \end{aligned}$$

$$\begin{aligned} &= \frac{b_{s-1}}{N} \left(\frac{m}{2} (N_+ + N_-) - \frac{N_+^2 + N_-^2}{2}\right) = \\ &= \frac{b_{s-1}}{2N} (mN - (N_+^2 + N_-^2 + 2N_+N_- - 2N_+N_-)) = \\ &= \frac{1}{2} b_{s-1} \left(m - \frac{(N_+ + N_-)^2}{N} + \frac{2N_+N_-}{N}\right) = \\ &= \frac{1}{2} b_{s-1} \left(m - N + 2N \left(\left(\frac{N_+}{N} \frac{N_-}{N}\right)^{\frac{1}{2}}\right)^2\right) \leq \\ &\leq \frac{1}{2} b_{s-1} \left(m - N + 2N \left(\frac{N_+}{N} + \frac{N_-}{N}\right) \frac{1}{4}\right) = \\ &= \frac{1}{2} b_{s-1} \left(m - N + \frac{1}{2} N\right) = \frac{1}{2} b_{s-1} \left(m - \frac{1}{2} N\right). \end{aligned} \quad (7)$$

Так как N является делителем функции Эйлера $\varphi(m) = m - 1$, то N представимо в виде $N = (m - 1)/d$, где d — делитель ЦЧ $\varphi(m) = m - 1$ ($d \neq N$). С учетом отмеченного обстоятельства из выражения (7) получаем

$$\begin{aligned} \max\{X^{(s)}\} - \min\{X^{(s)}\} &< \frac{1}{2} b_{s-1} \left(m - \frac{m-1}{2d}\right) = \\ &= \frac{1}{2} b_{s-1} m \left(1 - \frac{1}{2d} + \frac{1}{2dm}\right). \end{aligned}$$

Отсюда заключаем, что

$$\begin{aligned} \log_2(\max\{X^{(s)}\} - \min\{X^{(s)}\} + 1) + 1 &< \\ &< \log_2(b_{s-1}) + b_{\text{mod}} + \log_2\left(1 - \frac{1}{2d} + \frac{1}{2dm}\right). \end{aligned}$$

Таким образом, ввиду того, что $1/2 < 1 - 1/(2d) + 1/(2dm) < 1$ для разрядности ЦЧ $X^{(s)}$ справедлива оценка:

$$b_s = \lceil \log_2(\max\{X^{(s)}\} - \min\{X^{(s)}\} + 1) \rceil < b_{\text{mod}} + \log_2(b_{s-1}). \quad (8)$$

Что касается числа $X^{(1)}$ (см. (4)), то для оценки его разрядности b_1 также применим рассмотренный выше подход. Это обеспечивается тем, что при любом целочисленном C последовательность $\{W_0(C), W_1(C), \dots, W_{N-1}(C), W_N(C), W_{N+1}(C), \dots, W_{2N-1}(C), \dots, W_{b_0-1}(C)\}$ абсолютных наименьших остатков по модулю m , определяемых по правилу (3), благодаря выполнению равенств $W_{iN+j}(C) = W_j(C)$ ($i = \overline{1, \overline{[b_0/N]-1}}; j = \overline{0, N-1}$), как и последовательности (6), имеет циклическую структуру (с пе-

риодом N). Математические выкладки, приведенные выше для $X^{(s)}$ ($s = \overline{2, S}$), дают для разрядности b_1 числа $X^{(1)}$ оценку аналогичную (8), а именно оценку вида:

$$b_1 < b_{\text{mod}} + \log_2 b_0 = b_{\text{mod}} + \log_2 b. \quad (9)$$

Несмотря на то что оценочные значения (8), (9) разрядностей b_s ($s = \overline{1, S}$) чисел (4) сильно завышены, они вполне адекватно отражают характер схожимости рассматриваемой редукционной схемы.

Рекурсивный редукционный процесс, базирующийся на (2)—(4), обладает следующими свойствами.

А. Числа $X^{(0)} = X, |X^{(1)}|, |X^{(2)}|, \dots, |X^{(S)}|$ образуют убывающую последовательность. При этом ввиду (3) все $X^{(s)}$ ($s = \overline{0, S}$) равноостаточны по модулю m , т. е. являются элементами одного и того же класса \overline{X} вычетов по данному модулю: $\overline{X} = \{A \in \mathbf{Z} \mid A \equiv X \pmod{m}\}$ (\mathbf{Z} — множество целых чисел).

Б. Согласно (8), (9) достигаемое на s -й итерации отклонение $\Delta_s = b_s - b_{\text{mod}}$ разрядности ЦЧ $X^{(s)}$ от разрядности модуля m составляет порядка $[\log_2 b_s - 1]$ бит. Поскольку с увеличением s скорость приближения b_s к b_{mod} снижается, то в целях уменьшения числа S итераций вычислительной схемы (4), а значит и временных затрат на ее реализацию, в качестве признака завершения редукционного процесса принимается выполнение неравенства

$$\Delta_s = b_s - b_{\text{mod}} \leq \Delta_{\min}, \quad (10)$$

где Δ_{\min} — некоторый порог, подбираемый экспериментально (в ходе обучения соответствующей нейронной сети). В частности, при использовании в (10) $\Delta_{\min} = 0$ искомое значение выходной величины схемы (4) формируется по правилу:

$$\chi = \begin{cases} X^{(S)} + m, & \text{если } X^{(S)} < 0, \\ X^{(S)}, & \text{если } 0 \leq X^{(S)} < m, \\ X^{(S)} - m, & \text{если } m \leq X^{(S)}. \end{cases} \quad (11)$$

Число S итераций редукционной схемы (4) не превышает оценочного значения, определяемого условием

$$b_s - b_{\text{mod}} < \log_2(b_{\text{mod}} + \log_2(b_{\text{mod}} + \log_2(\dots \log_2(b_{\text{mod}} + \log_2 b_0) \dots))) \leq \Delta_{\min}(b_0 = b),$$

которое вытекает из (8)—(10).

В. Расчетные соотношения схемы (4) целиком согласуются с принципами нейросетевой вычислительной технологии. Осуществляя суммирова-

ние синаптических весов (3) с последующим вычислением активационной функции $\chi = |CX|_m = |X^{(s)}|_m$, реализуемой, например, в виде (11). Набор необходимых весовых коэффициентов (3) рассчитывается предварительно и хранится в памяти.

Г. Редукционная схема (4) без существенных структурно-функциональных изменений применима для любого числа модулей. Это и обеспечивает возможность совместного использования ИНС и МА.

Д. Метод модулярной редукции для преобразования $X \rightarrow |CX|_m$ по рекурсивной вычислительной схеме (4) легко может быть обобщен на случай использования произвольной позиционной системы счисления с основанием $r > 2$ и, в частности, десятичной системы счисления.

2. Редукционный алгоритм позиционно-модулярного кодового преобразования целых чисел

На базе представленного метода последовательного уменьшения разрядности ЦЧ по редукционной схеме (4) рекурсивного типа синтезирован алгоритм позиционно-модулярного кодового преобразования, ориентированный на нейросетевую реализацию, который заключается в нижеследующем.

Параметры алгоритма:

- попарно простые модули m_1, m_2, \dots, m_k , имеющие соответственно разрядности $b_{\text{mod}_1}, b_{\text{mod}_2}, \dots, b_{\text{mod}_k}$ бит ($b_{\text{mod}_i} = [\log_2 m_i]$ ($i = \overline{1, k}$); $k \geq 1$);
 - порог Δ_{\min} ($\Delta_{\min} = 0$) для решающего правила (10) завершения редукционного процесса.
- Входные данные алгоритма:
- двоичный код $(x_{b-1} x_{b-2} \dots x_0)_2$ исходного ЦЧ X (b — длина кода);
 - целочисленные коэффициенты C_i произведений $C_i X$, подлежащих приведению к остаткам по модулям m_i ($i = \overline{1, k}$).

Выходные данные: набор остатков — модулярный код $(\chi_1, \chi_2, \dots, \chi_k)$ ($\chi_i = |C_i X|_{m_i}$, $i = \overline{1, k}$) по заданному базису модулей — $\mathbf{M} = \{m_1, m_2, \dots, m_k\}$.

Предварительно получаемые данные: рассчитанные согласно правилу (3) наборы весов

$$\mathbf{W}_i(C) = \left\{ W_{j,i}(C) \mid W_{j,i}(C) = |C_i 2^j|_{m_i}^-; j = \overline{0, b-1} \right\}, \quad (12)$$

$$\mathbf{W}_i = \left\{ W_{j,i} \mid W_{j,i} = |2^j|_{m_i}^-; j = \overline{0, b_1-1}; b_1 < b \right\}, \quad (13)$$

$i = \overline{1, k}.$

В случае, когда базис \mathbf{M} содержит только один модуль ($\mathbf{M} = \{m\}$), индекс i в (12), (13) опускается.

Тело алгоритма позиционно-модулярного кодового преобразования по редукционной схеме понижения разрядности вычетов по модулю:

ПМ_РС.1. Положить $b_0 = b$, $X^{(0)} = (x_{b_0-1}^{(0)} x_{b_0-2}^{(0)} \dots x_0^{(0)})_2 = (x_{b-1} x_{b-2} \dots x_0)_2 = X$, $i = 1$.

ПМ_РС.2. Номеру итерации редукционного процесса присвоить начальные значения: $s = 1$.

ПМ_РС.3. Вычислить $X^{(1)} = \sum_{j=0}^{b_0-1} W_{j,i}(C_i) x_j^{(0)}$, сформировав двоичный код $(x_{b_1-1}^{(1)} x_{b_1-2}^{(1)} \dots x_0^{(1)})_2$ длины b_1 бит ЦЧ $X^{(1)}$.

ПМ_РС.4. Если $b_{\text{mod } i} < b_s$, то s инкрементировать ($s = s + 1$), найти

$$X^{(s)} = \sum_{j=0}^{b_{s-1}-2} W_{j,i} x_j^{(s-1)} - W_{b_{s-1}-1,i} x_{b_{s-1}-1}^{(s-1)},$$

получая код $(x_{b_s-1}^{(s)} x_{b_s-2}^{(s)} x_0^{(s)})_2$ длины b_s бит числа $X^{(s)}$ и данный шаг алгоритма (шаг ПМ_РС.4) повторить сначала.

ПМ_РС.5. Ввиду $b_s \leq b_{\text{mod } i}$ в соответствии с (11) для фиксации искомого значения i -й цифры формируемого МК выполнить действия:

ПМ_РС.5А. При $X^{(s)} < 0$ положить $\chi_i = X^{(s)} + m_i$ и перейти к ПМ_РС.6.

ПМ_РС.5Б. В случае $X^{(s)} \geq m_i$ положить $\chi_i = X^{(s)} - m_i$ и перейти к ПМ_РС.6.

ПМ_РС.5В. Выполнить операцию присвоения: $\chi_i = X^{(s)}$.

ПМ_РС.6. Если $i \neq k$, то переменную i инкрементировать ($i = i + 1$) и перейти к ПМ_РС.2.

ПМ_РС.7. Завершить работу алгоритма.

Приведем демонстрационные примеры.

Пример 1. Пусть требуется найти остаток от деления ЦЧ $X = 987\ 654\ 321$ на модуль $m = 13$.

В целях упрощения расчетов воспользуемся версией алгоритма ПМ_РС.1— ПМ_РС.7, ориентированной на десятичную систему счисления. Сформируем набор необходимых весов:

$$\begin{aligned} \mathbf{W} &= \left\{ W_j \mid W_j = \left| 10^j \right|_m^- ; j = \overline{0, b-1} \right\} = \\ &= \left\{ \left| 10^0 \right|_{13}^-, \left| 10^1 \right|_{13}^-, \dots, \left| 10^8 \right|_{13}^- \right\} = \\ &= \{1, -3, -4, -1, 3, 4, 1, -3, -4\}. \end{aligned}$$

Выполняя первую итерацию рекурсивной редукционной схемы (4), вычислим

$$X^{(1)} = \sum_{j=0}^{b_0-1} W_j x_j^{(0)} = \sum_{j=0}^{b-1} W_j x_j.$$

Ввиду того, что

$$\begin{aligned} b_0 &= b = 9, X^{(0)} = (x_8^{(0)} x_7^{(0)} \dots x_0^{(0)})_{10} = \\ &= (x_8 x_7 \dots x_0)_{10} = 987\ 654\ 321 \end{aligned}$$

имеем:

$$\begin{aligned} X^{(1)} &= 1 \cdot 1 + (-3 \cdot 2) + (-4 \cdot 3) + \\ &+ (-1 \cdot 4) + 3 \cdot 5 + 4 \cdot 6 + 1 \cdot 7 + (-3 \cdot 8) + (-4 \cdot 9) = \\ &= 1 - 6 - 12 - 4 + 15 + 24 + 7 - 24 - 36 = -35. \end{aligned}$$

Разрядность дополнительного двоичного кода числа $X^{(1)} = -35$ составляет $\lceil \log_2 35 \rceil + 1 = 7$ бит, в то время как модуль $m = 13$ является четырехразрядным ($b_{\text{mod}} = 4$ бита). Поэтому редукционный процесс (4) должен быть продолжен. Вторая его итерация дает:

$$X^{(2)} = -(5 + (-3 \cdot 3)) = 4.$$

Так как $0 < X^{(2)} < m = 13$, то $X^{(2)}$ — искомый остаток. Число итераций редукционного процесса составляет $S = 2$.

Пример 2. Пусть в МСС с базисом $\{m_1, m_2, m_3\} = \{5, 11, 13\}$ требуется найти нормированный остаток $\chi_{3,3} = \left| \mu_{3,3} \chi_3 \right|_{m_3}$ для вычета $\chi_3 = 11$ по модулю $m_3 = 13$, где $\mu_{3,3} = \left| M_{3,3}^{-1} \right|_{m_3} = \left| (m_1 m_2)^{-1} \right|_{m_3} = \left| 1 / (5 \cdot 11) \right|_{13} = \left| 1/3 \right|_{13} = 9$.

В двоичной системе счисления вычет $\chi_3 = 11$ представим в виде $\chi_3 = (x_3 x_2 x_1 x_0)_2 = (1011)_2 = 2^0 \cdot 1 + 2^1 \cdot 1 + 2^2 \cdot 0 + 2^3 \cdot 1$. Следуя принятым обозначениям, положим $m = m_3 = 13$, $X = X^{(0)} = \chi_3 = 11$, $b_0 = b = 4$ бита, $C = \mu_{3,3} = 9$, в соответствии с (3) сформируем используемый на первой (начальной) итерации алгоритма ПМ_РС.1— ПМ_РС.7 набор коэффициентов для вычисления сформированного остатка $\chi_{3,3} = |CX|_m = |9 \cdot 11|_{13}$:

$$\begin{aligned} \mathbf{W}(C) &= \{W_0(9), W_1(9), W_2(9), W_3(9)\} = \\ &= \left\{ \left| 9 \cdot 2^0 \right|_{13}^-, \left| 9 \cdot 2^1 \right|_{13}^-, \left| 9 \cdot 2^2 \right|_{13}^-, \left| 9 \cdot 2^3 \right|_{13}^- \right\} = \\ &= \{9 \cdot 1|_{13}^-, 9 \cdot 2|_{13}^-, 9 \cdot 4|_{13}^-, 9 \cdot 8|_{13}^- \} = \{-4, 5, -3, -6\} \end{aligned}$$

На первой итерации реализуемого редукционного процесса получаем

$$\begin{aligned} X^{(1)} &= \sum_{j=0}^{b_0-1} W_j(C) x_j^{(0)} = \sum_{j=0}^{b-1} W_j(C) x_j = \\ &= (-4 \cdot 1) + 5 \cdot 1 + (-3 \cdot 0) + (-6 \cdot 1) = -5. \end{aligned}$$

Разрядность абсолютной величины $|X^{(1)}|$ ЦЧ $X^{(1)} = -5$ составляет $b_1 = 3$ бита, а модуль $m = 13$ является четырехбитовым ($b_{\text{mod}} = 4$ бита). Следовательно, $X^{(1)} \in \mathbf{Z}_{13}^- = \{-6, -5, \dots, 6\}$. Таким образом, выполняемый редукционный процесс состоит из одной итерации ($S = 1$). При этом в соответствии с (11) искомый нормированный

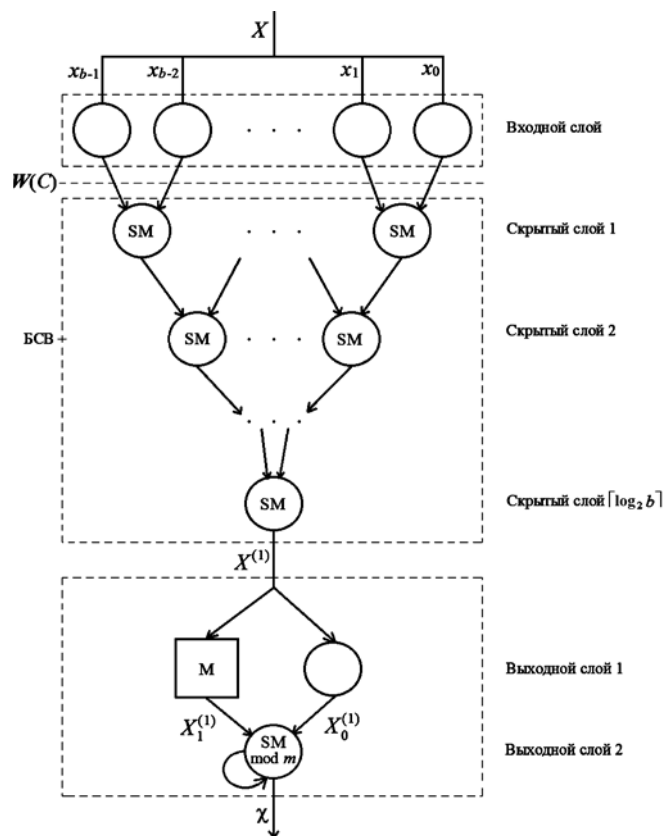
остаток $\chi_{3,3} = X^{(1)} + m = -5 + 13 = 8$, что совпадает с результатом непосредственных вычислений: $\chi_{3,3} = |CX|_m = |9 \cdot 11|_{13} = 8$.

3. Высокоскоростная нейросетевая реализация редуccionного алгоритма позиционно-модулярного преобразования больших чисел

Синтезированная процедура ПМ_РС.1—ПМ_РС.7 приведения взвешенных целых чисел к остаткам по модулям $m \in \mathbf{M} = \{m_1, m_2, \dots, m_k\}$ ($k \geq 1$) может быть реализована как программно, так и аппаратным способом с применением нейросетевой вычислительной технологии [1, 3, 18, 19].

На рисунке представлена структура быстродействующей параллельной НСКК, которая выполняет редуccionную схему (4) преобразования $X \rightarrow |CX|_m$ за одну итерацию ($S = 1$). Данная нейронная сеть включает входной слой, нейроны которого образуют b -разрядный регистр для фиксации двоичного кода $(x_{b-1} x_{b-2} \dots x_0)_2$ ЦЧ X , $\lceil \log_2 b \rceil$ скрытых слоев, в совокупности составляющих блок суммирования вычетов (БСВ) — взвешенных компонент набора вида (12):

$$\mathbf{W}(C) = \left\{ W_j(C) \mid W_j(C) = |C2^j|_m; j = \overline{0, b-1} \right\}, \quad (14)$$



Параллельная нейронная сеть конечного кольца для позиционно-модулярного преобразования по одноитерационной редуccionной схеме

а также два выходных слоя, осуществляющих приведение числа $X^{(1)}$ (см. (4)), получаемого блоком суммирования вычетов к остатку по модулю m .

В скрытых слоях используются сумматоры SM , которые выполняют операции сложения пар вычетов, формируемых в соответствующих предыдущих слоях. Если в l -й слой БСВ ($l = \overline{1, \lceil \log_2 b \rceil}$) поступает нечетное число N_l вычетов, то вычет, не вошедший в пару (условимся считать, что он имеет порядковый номер N_l) хранится в регистре в течение времени сложения пар вычетов в данном слое. Таким образом, БСВ имеет параллельную древовидную (пирамидальную) архитектуру конвейерного типа. Это обеспечивает получение на выходе БСВ двоичного кода ЦЧ

$$\begin{aligned} X^{(1)} &= \sum_{j=0}^{b_1-1} W_j(C) x_j = \sum_{j=0}^{b_1-2} 2^j x_j^{(1)} - 2^{b_1-1} x_{b_1-1}^{(1)} = \\ &= \sum_{j=0}^{b_1-2} W_j x_j^{(1)} - W_{b_1-1} x_{b_1-1}^{(1)} \end{aligned} \quad (15)$$

(см. (4)) за время $\lceil \log_2 b \rceil t_{\text{сл}}$, где $t_{\text{сл}}$ — длительность операции сложения двух вычетов.

Поскольку в наборе (14) могут быть как положительные, так и отрицательные компоненты, то их суммирование должно проводиться в дополнительном двоичном коде. Согласно оценке (9) разрядность b_1 кода $((x_{b_1-1}^{(1)} x_{b_1-2}^{(1)} \dots x_0^{(1)})_2$ ЦЧ $X^{(1)}$ удовлетворяет неравенству

$$\begin{aligned} b_1 &\leq \Delta + b_{\text{mod}} \quad (\Delta = \lceil \log_2 b \rceil; \\ &b_{\text{mod}} = \lceil \log_2 m \rceil). \end{aligned} \quad (16)$$

Пусть в (16) достигается равенство. Тогда из (15) вытекает соотношение

$$\left| X^{(1)} \right|_m = \left| X_0^{(1)} + X_1^{(1)} \right|_m, \quad (17)$$

где

$$X_0^{(1)} = \sum_{j=0}^{b_{\text{mod}}-2} 2^j x_j^{(1)} = \sum_{j=0}^{b_{\text{mod}}-2} W_j x_j^{(1)}; \quad (18)$$

$$\begin{aligned} X_1^{(1)} &= \left| \sum_{j=b_{\text{mod}}-1}^{\Delta+b_{\text{mod}}-2} 2^j x_j^{(1)} - 2^{\Delta+b_{\text{mod}}-1} x_{\Delta+b_{\text{mod}}-1}^{(1)} \right|_m = \\ &= \left| \sum_{j=b_{\text{mod}}-1}^{\Delta+b_{\text{mod}}-2} W_j x_j^{(1)} - W_{\Delta+b_{\text{mod}}-1} x_{\Delta+b_{\text{mod}}-1}^{(1)} \right|_m. \end{aligned} \quad (19)$$

Вычеты W_j ($j = \overline{0, \Delta + b_{\text{mod}} - 1}$) определяют по (3), (13). Равенства (17)—(19) положены в основу блока приведения ЦЧ $X^{(1)}$ к результирующему остатку $\chi = |X^{(1)}|_m$ по модулю m . Значения b_{mod} — битового вычета $X_1^{(1)}$ по модулю m рассчитываются предварительно и записываются в

табличную память M по соответствующим адресам:

$$\left(x_{\Delta+b_mod-1}^{(1)} x_{\Delta+b_mod-2}^{(1)} \dots x_{b_mod-1}^{(1)}\right)_2,$$

имеющим разрядность $\Delta + 1$ бит. Таким образом, емкость необходимой табличной памяти составляет $2^{\Delta+1} \times b_mod$ бит.

Согласно вышеизложенному НСКК, представленная на рисунке, работает следующим образом. Двоичный код $(x_{b-1} x_{b-2} \dots x_0)_2$ ЦЧ X , подлежащего преобразованию $X \rightarrow |CX|_m$, фиксируется в нейронах входного слоя, откуда вместе с набором $\mathbf{W}(C)$ синаптических весов поступает в БСВ. Реализуя рекурсивную $\lceil \log_2 b \rceil$ -каскадную процедуру суммирования аддитивных компонент соотношения (15) с весами набора (14) в режиме максимального распараллеливания вычислительного процесса на уровне двухместных операций сложения в скрытых слоях с первого по $\lceil \log_2 b \rceil$ -й, БСВ получает дополнительный двоичный код $(x_{\Delta+b_mod-1}^{(1)} x_{\Delta+b_mod-2}^{(1)} \dots x_0^{(1)})_2$ числа $X^{(1)}$. Младшая $b_mod - 1$ -битовая часть $X_0^{(1)} = (x_{b_mod-2}^{(1)} x_{b_mod-3}^{(1)} \dots x_0^{(1)})_2$ сформированного кода сохраняется в регистре первого выходного слоя, а старшая часть $(x_{\Delta+b_mod-1}^{(1)} x_{\Delta+b_mod-2}^{(1)} \dots x_{b_mod-1}^{(1)})_2$ подается на адресный вход таблицы M и из нее извлекается остаток $X_1^{(1)}$ от деления старшей части ЦЧ $X^{(1)}$ на m . Вычеты $X_0^{(1)}$ и $X_1^{(1)}$ поступают во второй выходной слой, где параллельный сумматор $SM \bmod m$ по модулю m с обратной связью выполняет заключительную операцию сложения по модулю m : $\chi = |CX|_m = |X^{(1)}|_m = |X_0^{(1)} + X_1^{(1)}|_m$. В общей сложности редуцирующая процедура преобразования $X \rightarrow |CX|_m$ параллельной НСКК со структурой, приведенной на рисунке, осуществляется за время порядка $(\lceil \log_2 b \rceil + 2)t_{сл}$.

В процессе разработки нейрокомпьютерного обеспечения современных МА-приложений, в том числе криптографических, НСКК играют ключевую роль.

Заключение

Основные результаты представленных в работе прикладных исследований по проблематике создания НСКК для высокопроизводительных МА-приложений в области защиты информации состоят в нижеследующем

1. Для построения нейронных сетей на конечных кольцах вычетов по модулям МСС, как основы нейросетевых модулярных вычислительных струк-

тур криптографического назначения, принята редуцирующая технология позиционно-модулярного преобразования больших чисел с расширенными функциональными возможностями. Обеспечивая оптимальные условия для согласования и реализации фундаментальных свойств параллелизма НС и МА, в рамках развиваемых подходов к решению поставленной задачи применяется также табличный метод ускорения выполняемого рекурсивного процесса поитерационного понижения разрядности формируемых двоичных кодов.

2. Дана математическая формализация редуцирующего метода позиционно-модулярного кодового преобразования. Получены оценки диапазона изменения, а также разрядности элементов последовательности поитерационных вычетов и на этой основе проведено исследование характера и скорости сходимости применяемой редуцирующей схемы, предложены эффективные способы сокращения числа итераций выполняемого преобразования.

3. Синтезирован общий алгоритм преобразования взвешенных больших чисел из двоичного в модулярный код по редуцирующей схеме понижения разрядности элементов формируемой последовательности вычетов. В качестве признака завершения реализуемого рекурсивного процесса по модулям МСС используется фиксация вычетов, разрядности которых достигают значения, отличающегося от разрядностей соответствующих модулей на величину, не превышающую установленный порог.

4. Разработана параллельная структура НСКК, осуществляющей позиционно-модулярное кодовое преобразование по редуцирующей схеме за одну итерацию — за время порядка $(\lceil \log_2 b \rceil + 2)t_{сл}$ (b — разрядность входного числа; $t_{сл}$ — длительность операции сложения двух вычетов).

Список литературы

1. Червяков Н. И., Коляда А. А., Ляхов П. А. и др. Модулярная арифметика и ее приложения в инфокоммуникационных технологиях. М.: Физматлит, 2017. 400 с.
2. Ananda Mohan P. V. Residue number systems: Theory and applications. Basel: Birkhauser, Mathematics, 2016. 351 p.
3. Червяков Н. И. Применение искусственных нейронных сетей и системы остаточных классов в криптографии. М.: Физматлит, 2012. 280 с.
4. Инютин С. А. Основы модулярной алгоритмики. Ханты-Мансийск: Полиграфист, 2009. 347 с.
5. Amos O., Premkumar B. Residue number systems: Theory and implementation. Singapore: Imperial college press, 2007. 311 p.
6. Оцков Ш. А. Способ организации высокоточных вычислений в модулярной арифметике // Первая международная конференция "Параллельная компьютерная алгебра и ее приложения в новых инфокоммуникационных

системах". Ставрополь, РФ, 20—24 окт., 2014: сборник науч. трудов. Ставрополь: Фабула, 2014. С. 270—277.

7. **Комарова Ю. А., Талалаев И. А.** Аналитический обзор методов и структур для работы с большими данными // Первая международная конференция "Параллельная компьютерная алгебра и ее приложения в новых инфокоммуникационных системах". Ставрополь, РФ, 20—24 окт., 2014: сборник науч. трудов. Ставрополь: Фабула, 2014. С. 477—485.

8. **Афонин М. С.** Способ обработки больших чисел на ПЛИС с малой ресурсной мощностью // Первая международная конференция "Параллельная компьютерная алгебра и ее приложения в новых инфокоммуникационных системах". Ставрополь, РФ, 20—24 окт., 2014: Сборник науч. трудов. Ставрополь: ИИЦ "Фабула", 2014. С. 511—520.

9. **Gulang G.-B., Mao K.-Z., Siew C. K., Huang D.-S.** Fast modular network implementation support vector machines // IEEE Trans. Neural Networks. 2005. Vol. 16, N. 6. P.1651—1663.

10. **Тихонов Э. Е., Евдокимов А. А.** Программно-аппаратная реализация нейронных сетей. Невинномысск: НИЭУП, 2013. 116 с.

11. **Daniela S., Melin P., Castillo O.** Optimization of modular granular neural networks using arhierarchical genetic algorithm based on the database comlexcity applied to human recognition // Informations Sciences. Tjuana Institute of Technology. 2015. Vol. 309. Tjuana, Mexico. P. 73—101.

12. **Кондрашев А. В., Горденко Д. В., Павлюк Д. Н.** Нейронная сеть для преобразования чисел, представленных

в позиционном коде, в систему остаточных классов // Исследования в области естественных наук. 2015. № 1. URL: <http://science.snauka.ru/2015/01/8925>

13. **Бабенко М. Г., Черных А. Н., Кучуков В. А., Дерябин М. А., Кучукова Н. Н.** Разработка нового нейросетевого метода вычисления модульного умножения в системе остаточных классов // Нейрокомпьютеры: разработка и применение. 2016. № 10. С. 41—48.

14. **Коляда А. А.** Обобщенная интегрально-характеристическая база модулярных систем счисления // Информационные технологии. 2017. Т. 23, № 9. С. 641—649.

15. **Чернявский А. Ф., Коляда А. А.** Преобразование кода модулярной системы счисления в обобщенный позиционный код // Доклады НАН Беларуси. 2017. Т. 61, № 4. С. 26—30.

16. **Виноградов И. М.** Основы теории чисел. СПб.: Лань, 2009. 176 с.

17. **Корн Г.** Справочник по математике для научных работников и инженеров. М.: Наука, 1973. 831 с.

18. **Червяков Н. И., Евдокимов А. А.** Нейронные сети конечного кольца для реализации пороговых схем разделения секрета // Нейрокомпьютеры: Разраб., применение. 2007. № 2—3. С. 45—50.

19. **Червяков Н. И., Спельников А. Б., Мезенцева А. Ф.** Нейронная сеть конечного кольца прямого распространения для операций на эллиптических кривых // Нейрокомпьютеры: Разраб., применение. 2008. № 1—2. С. 28—34.

A. A. Kolyada, D. Sc., Associate Professor; E-mail: razan@tut.by,

P. V. Kuchynski, D. Sc., Associate Professor Director of IAPP "Institute of Applied Physics Problems of A. N. Sevchenko" Belarusian State University (IAPP of A. N. Sevchenko), Minsk, 220045, Belarus, Kurchatov St., 7; E-mail: niipfp@bsu.by,

N. I. Chervyakov, D. Sc., Professor, Head of Department, Federal State Autonomous Educational Institution of Higher Professional Education "North-Caucasus Federal University", Stavropol, 355029, Russian Federation; E-mail: Chervyakov@yandex.ru, whbear@yandex.ru

Reducing Method of Positional-Modular Converting Large Numbers for Neural Networks to the End Rings

The article is devoted to the problem of constructing neural networks of a finite ring (NNFR), which serve as the basis for neural network modular computing structures for high-performance cryptographic applications. The methodological basis of the NNFR of the investigated class is the modified reduction method of position-modular transformation of weighted large numbers. The authors give a mathematical formalization of the method, was obtained estimates for the range of change and the number of elements of the residue sequence formed by the reduction scheme of the recursive type, the nature and speed of its convergence are investigated, a flexible tabular mechanism for reducing the number of iterations of the scheme is proposed. Synthesized general reducing algorithm of position-modular code conversion, was developed the parallel structure of the NNFR which performs a basic transformation in one iteration — during the time of order $(\lceil \log_2 b \rceil + 2)t_{ca}$, where b is the number of the input number; t_{ca} is the duration of the two-fold addition operation.

Keywords: neural network, neural network of finite ring, synaptic weights, modular number system, modular arithmetic, cryptography, range of large numbers, reduction method of position-modular transformation

References

1. **Chervjakov N. I., Koljada A. A., Ljahov P. A.** et al. *Moduljarnaja arifmetika i ee prilozhenija v infokommunikacionnyh tehnologijah* [Modular arithmetic and its applications in infocommunication technologies], Moscow, Fizmatlit Publ., 2017, 400 p. (in Russian).
2. **Ananda Mohan P. V.** *Residue number systems: Theory and applications*, Basel, Birkhauser, Mathematics, 2016, 351 p.
3. **Chervjakov N. I.** *Primenenie iskusstvennyh neyronnyh setej i sistemy ostatocnyh klassov v kriptografii* [The use of artificial neural networks and the residual class system in cryptography], Moscow, Fizmatlit Publ., 2012, 280 p. (in Russian).
4. **Injutin S. A.** *Osnovy moduljarnoj algoritmiki* [Fundamentals of modular algorithms], Khanty-Mansiysk, Poligrafist Publ., 2009, 347 p. (in Russian).
5. **Amos O., Premkumar B.** *Residue number systems: Theory and implementation*. Singapore, Imperial college press, 2007, 311 p.
6. **Ocovok Sh. A.** The way to organize high-precision calculations in modular arithmetic, *Pervaja mezhdunarodnaja konferencija "Parallel'naja komp'juternaja algebra i ee prilozhenija v novyh infokommunikacionnyh sistemah"* [First International Conference "Parallel Computer Algebra and Its Applications in New Infocommunication Systems"], Stavropol, Russian Federation, 20—24 okt., 2014, Fabula Publ., 2014, pp. 270—277. (in Russian).
7. **Komarova Ju. A., Talalaev I. A.** Analytical review of methods and structures for working with large data, *Pervaja mezhdunarodnaja konferencija "Parallel'naja komp'juternaja algebra i ee prilozhenija v novyh infokommunikacionnyh sistemah"* [First International Conference "Parallel Computer Algebra and Its Applications in New Infocommunication Systems"], Stavropol, Russian Federation, 20—24 okt., 2014, Fabula Publ., 2014, pp. 477—485 (in Russian).
8. **Afonin M. S.** The way of processing large numbers on a FPGA with a small resource capacity, *Pervaja mezhdunarodnaja konferencija "Parallel'naja komp'juternaja algebra i ee prilozhenija v novyh infokommunikacionnyh sistemah"* [First International Conference "Parallel Computer Algebra and Its Applications in New Infocommunication Systems"], Stavropol, Russian Federation, 20—24 okt., 2014, Fabula Publ., 2014, pp. 511—520 (in Russian).
9. **Gulang G.-B., Mao K.-Z., Siew C. K., Huang D.-S.** Fast modular network implementation support vector machines, *IEEE Trans. Neural Networks*, 2005, vol. 16, no. 6, pp. 1651—1663.
10. **Tihonov Je. E., Evdokimov A. A.** *Programmno-apparatnaja realizacija neyronnyh setej* [Software and hardware implementation of neural networks: Monograph], Nevinnomyssk, NIJeUP Publ., 2013, 116 p. (in Russian).
11. **Daniela S., Melin P., Castillo O.** Optimization of modular granular neural networks using arhierarchical genetic algorithm based on the database complexcity applied to human recognition, *Informations Sciences. Tjuana Institute of Technology*, 2015, Vol. 309, Tjuana, Mexico, pp. 73—101.
12. **Kondrashjov A. V., Gordenko D. V., Pavljuk D. N.** Neural network for converting the numbers represented in the positional code to the residual class system. *Issledovanija v oblasti estestvennyh nauk* [Research in the field of natural sciences], 2015, no. 1, available at: <http://science.snauka.ru/2015/01/8925> (in Russian).
13. **Babenko M. G., Chernyh A. N., Kuchukov V. A., Derjabin M. A., Kuchukova N. N.** Development of a new neural network method for calculating modular multiplication in a system of residual classes, *Nejrokomputery: razrabotka i primenenie* [Neurocomputers: development and application], 2016, no. 10, pp. 41—48 (in Russian).
14. **Koljada A. A.** Generalized integral-characteristic base of modular number systems, *Informacionnye tehnologii* [Information Technology] 2017, vol. 23, no. 9, pp. 641—649 (in Russian).
15. **Chernjavskij A. F., Koljada A. A.** Conversion of a modular number system code to a generalized positional code, *Doklady NAN Belarusi* [Reports of NASc of Belarus], 2017, vol. 61, no. 4, pp. 26—30 (in Russian).
16. **Vinogradov I. M.** *Osnovy teorii chisel* [Fundamentals of number theory], St. Petersburg, Lan' Publ., 2009, 176 p. (in Russian).
17. **Korn G.** *Spravochnik po matematike dlja nauchnyh rabotnikov i inzhenerov* [A handbook on mathematics for scientists and engineers], Moscow, Nauka Publ., 1973, 831 p. (in Russian).
18. **Chervjakov N. I., Evdokimov A. A.** Neural networks of the finite ring for the implementation of threshold separation schemes for secretion, *Nejrokomputery: Razrab., primenenie* [Neurocomputers: Development, application], 2007, no. 2—3, pp. 45—50 (in Russian).
19. **Chervjakov N. I., Spel'nikov A. B., Mezenceva A. F.** Neural network of a finite ring of direct propagation for operations on elliptic curves, *Nejrokomputery: Razrab., primenenie* [Neurocomputers: Development, application], 2008, no. 1—2, pp. 28—34 (in Russian).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова.*

Корректор *Е. В. Комиссарова.*

Сдано в набор 07.03.2018. Подписано в печать 26.04.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT518. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.

ПУМСС-2018

3-6 сентября
2018 года

Самара, Россия



ПРОБЛЕМЫ УПРАВЛЕНИЯ И МОДЕЛИРОВАНИЯ В СЛОЖНЫХ СИСТЕМАХ

XX Международная конференция

Первое оповещение

Уважаемые коллеги!

Федеральное государственное бюджетное учреждение науки Институт проблем управления сложными системами Российской академии наук в период с 3 по 6 сентября 2018 года проводит в городе Самаре очередную XX Международную научную конференцию «Проблемы управления и моделирования в сложных системах».

Целью конференции является повышение эффективности научных исследований в области решения проблем управления и моделирования в сложных системах путём обсуждения последних научных достижений и лучших практик их применения в различных областях.

Направления конференции:

- Методы управления и оптимизации в сложных технических системах
- Интеллектуальные технологии в сложных системах
- Измерения, контроль и диагностика в экстремальных условиях
- Процессы управления в обществе
- Цифровое сельское хозяйство и агрокибернетика
- Биокибернетика и биоинформатика

Формы участия:

- Обзорные и обобщающие доклады (30 минут)
- Доклады, посвящённые частным вопросам (10 минут)

Языки конференции:

- русский
- английский

Важные даты

25 марта	Завершение приема заявок на участие
15 апреля	Завершение работы по отбору докладов
10 мая	Завершение приема электронных копий рукописей докладов
1 июня	Завершение приема твердых копий рукописей докладов с сопровождающими документами
30 июня	Завершение приема оплаты организационного взноса

Публикации:

По итогам конференции издается сборник трудов, индексируемый в РИНЦ.

По решению программного комитета расширенные версии отдельных докладов будут рекомендованы для опубликования в журналах из Перечня ВАК:

- Вестник СамГТУ. Серия "Технические науки"
- Онтология проектирования
- Инфокоммуникационные технологии

Дополнительная информация

ИПУСС РАН

ул. Садовая, 61, г. Самара, 443020

тел. +7 (846) 333 27 70

<http://www.iccs.ru/cscmp/cscmp.html>

e-mail: cscmp@iccs.ru