

Software & Systems
Programmnye produkty i sistemy
2017, vol. 30, no. 2

Print ISSN 0236-235X

Online ISSN 2311-2735

Программные продукты и системы

DOI 10.15827/0236-235X

/ Методы
комбинаторного
моделирования

/ Алгоритм
конструирования
ассоциативных
правил

/ Реализация
web-инструментария
распознавания образов

/ Автоматизированная
система обработки
данных



№ 2
ТОМ 30
2017



Научно-исследовательский институт
«Центрпрограммсистем»

Программные продукты и системы

МЕЖДУНАРОДНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

2017, том 30, № 2
(год издания тридцатый)

Главный редактор
С.В. ЕМЕЛЬЯНОВ, академик РАН

Тверь

SOFTWARE & SYSTEMS

(PROGRAMMNYE PRODUKTY I SISTEMY)

International research and practice journal

2017, vol. 30, no. 2

Editor-in-Chief
S.V. EMEL'YANOV, Academician of the Russian Academy of Sciences

Tver
Russian Federation

Research Institute CENTERPROGRAMSYSTEM

© ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

Международный научно-практический журнал

2017. Т. 30. № 2

DOI: 10.15827/0236-235X.118

Главный редактор

С.В. ЕМЕЛЬЯНОВ,

академик РАН (г. Москва, Россия)

Научные редакторы:

А.П. АФАНАСЬЕВ, д.ф.-м.н., профессор МФТИ (ТУ)

(г. Москва, Россия)

Н.А. СЕМЕНОВ, д.т.н., профессор ТвГТУ

(г. Тверь, Россия)

В.М. КУРЕЙЧИК, д.т.н., профессор ИТА ЮФУ

(г. Таганрог, Россия)

Издатель НИИ «Центрпрограммсистем»

(г. Тверь, Россия)

Учредители: МНИИПУ (г. Москва, Россия),

Главная редакция международного журнала

«Проблемы теории и практики управления» (г. Москва, Россия),

Закрытое акционерное общество

«Научно-исследовательский институт

«Центрпрограммсистем» (г. Тверь, Россия)

Журнал зарегистрирован в Комитете Российской Федерации

по печати 26 июня 1995 г.

Регистрационное свидетельство № 013831

Подписной индекс в каталоге

Агентства «Роспечать» **70799**

ISSN 0236-235X (печатн.)

ISSN 2311-2735 (онлайн)

МЕЖДУНАРОДНАЯ РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Семенов Н.А. – д.т.н., профессор Тверского государственного технического университета, заместитель главного редактора (г. Тверь, Россия)

Решетников В.Н. – д.ф.-м.н., профессор Московского авиационного института (национального исследовательского университета), заместитель главного редактора (г. Москва, Россия)

Арефьев И.Б. – д.т.н., профессор Морской академии Польши (г. Щецин, Польша)

Афанасьев А.П. – д.ф.-м.н., профессор Московского физико-технического института (технического университета), заведующий Центром распределенных вычислений Института проблем передачи информации РАН (г. Москва, Россия)

Баламетов А.Б. – д.т.н., профессор Азербайджанского научно-исследовательского и проектно-изыскательского института энергетики (г. Баку, Азербайджан)

Батыршин И.З. – д.т.н., профессор Мексиканского института нефти (г. Мехико, Мексика)

Вагин В.Н. – д.т.н., профессор Московского энергетического института (технического университета)

(г. Москва, Россия)

Голенков В.В. – д.т.н., профессор Белорусского государственного университета информатики и радиоэлектроники

(г. Минск, Беларусь)

Еремеев А.П. – д.т.н., профессор Московского энергетического института (технического университета)

(г. Москва, Россия)

Котов А.С. – кандидат наук, ассистент профессора университета Уэйна (штат Мичиган) (г. Детройт, США)

Кузнецов О.П. – д.т.н., профессор Института проблем управления РАН (г. Москва, Россия)

Курейчик В.М. – д.т.н., профессор Инженерно-технологической академии Южного федерального университета

(г. Таганрог, Россия)

Лисецкий Ю.М. – к.т.н., генеральный директор «S&T Ukraine» (г. Киев, Украина)

Мамросенко К.А. – к.т.н., доцент Московского авиационного института

(национального исследовательского университета), руководитель Центра визуализации и спутниковых

информационных технологий НИИСИ РАН (г. Москва, Россия)

Мейер Б. – доктор наук, профессор, заведующий кафедрой Высшей политехнической школы – ЕТН (г. Цюрих, Швейцария)

Нузен Тхань Нгу – д.ф.-м.н., профессор, проректор Ханойского открытого университета (г. Ханой, Вьетнам)

Николов Р.В. – доктор наук, профессор Университета библиотековедения и информационных технологий Софии

(г. София, Болгария)

Осипов Г.С. – д.ф.-м.н., профессор, заместитель директора Института системного анализа РАН (г. Москва, Россия)

Палюх Б.В. – д.т.н., профессор Тверского государственного технического университета (г. Тверь, Россия)

Рахманов А.А. – д.т.н., профессор, заместитель генерального директора Концерна «РТИ Системы» (г. Москва, Россия)

Серов В.С. – д.ф.-м.н., профессор Университета прикладных наук Оулу (г. Оулу, Финляндия)

Сотников А.Н. – д.ф.-м.н., профессор, Межведомственный суперкомпьютерный центр РАН (г. Москва, Россия)

Сулейманов Д.Ш. – академик АН Республики Татарстан, д.т.н., профессор Казанского государственного

технического университета (г. Казань, Республика Татарстан, Россия)

Тарасов В.Б. – к.т.н., доцент Московского государственного технического университета им. Н.Э. Баумана (г. Москва, Россия)

Таратухин В.В. – доктор философии, управляющий директор Европейского исследовательского центра

в области информационных систем (ERCIS) Вестфальского университета им. Вильгельма (г. Мюнстер, Германия)

Хорошевский В.Ф. – д.т.н., профессор Московского физико-технического института (технического университета)

(г. Москва, Россия)

Язенин А.В. – д.ф.-м.н., профессор Тверского государственного университета (г. Тверь, Россия)

АССОЦИИРОВАННЫЕ ЧЛЕНЫ РЕДАКЦИИ

Московский энергетический институт (технический университет), г. Москва, Россия

Технологический институт Южного федерального университета, г. Таганрог, Россия

Тверской государственный технический университет, г. Тверь, Россия

Научно-исследовательский институт «Центрпрограммсистем», г. Тверь, Россия

АДРЕС РЕДАКЦИИ

Россия, 170024, г. Тверь, пр. 50 лет Октября, 3а

Телефон (482-2) 39-91-49

Факс (482-2) 39-91-00

E-mail: red@cps.tver.ru

www.swsys.ru

Подписано в печать 23.05.2017 г.

Отпечатано ООО ИПП «Фактор и К»

Россия, 170028, г. Тверь, ул. Лукина, д. 4, стр. 1

Выпускается один раз в квартал.

Год издания тридцатый. Формат 60×84 1/8. Объем 180 стр.

Заказ № 12. Тираж 1000 экз. Цена 257,40 руб.

Автор статьи отвечает за подбор, оригинальность и точность приводимого фактического материала.
Авторские гонорары не выплачиваются. При перепечатке материалов ссылка на журнал обязательна.

Editor-in-chief

S.V. Emelyanov, Academician of the Russian Academy of Sciences
(Moscow, Russian Federation)

Science editors:

A.P. Afanasiev, Dr.Sc. (Physics and Mathematics), Professor MIPT
(Moscow, Russian Federation)

N.A. Semenov, Dr.Sc. (Engineering), Professor TvSTU
(Tver, Russian Federation)

V.M. Kureichik, Dr.Sc. (Engineering), Professor AET SFU
(Taganrog, Russian Federation)

INTERNATIONAL EDITORIAL BOARD

Semenov N.A. – Dr.Sc. (Engineering), Professor of Tver State Technical University, Deputy Editor-in-Chief
(Tver, Russian Federation)

Reshetnikov V.N. – Dr.Sc. (Physics and Mathematics), Professor of Moscow Aviation Institute (National Research University),
Deputy Editor-in-Chief (Moscow, Russian Federation)

Arefev I.B. – Dr.Sc. (Engineering), Professor of Poland Szczecin Maritime Academy (Szczecin, Poland)

Afanasiev A.P. – Dr.Sc. (Physics and Mathematics), Professor of Moscow Institute of Physics and Technology,
Head of Centre for Distributed Computing of Institute for Information Transmission Problems (Moscow, Russian Federation)

Balametov A.B. – Azerbaijan Scientific-Research & Design-Prospecting Power Engineering Institute (Baku, Azerbaijan)

Batyrshtin I.Z. – Dr.Sc. (Engineering), Professor of Mexican Petroleum Institute (Mexico City, Mexico)

Vagin V.N. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Golenkov V.V. – Dr.Sc. (Engineering), Professor of Belarusian State University of Informatics and Radioelectronics
(Minsk, Republic of Belarus)

Eremeev A.P. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Kotov A.S. – Ph.D. (Computer Science), Assistant Professor, Wayne State University (Detroit, MI, USA)

Kuznetsov O.P. – Dr.Sc. (Engineering), Professor of the Institute of Control Sciences of the Russian Academy of Sciences
(Moscow, Russian Federation)

Kureichik V.M. – Dr.Sc. (Engineering), Professor of Academy of Engineering and Technology Southern Federal University
(Taganrog, Russian Federation)

Lisetskiy Yu.M. – Ph.D.Tech.Sc., CEO of "S&T Ukraine" (Kiev, Ukraine)

Mamrosenko K.A. – Ph.D. (Engineering), Associate Professor of Moscow Aviation Institute (National Research University),
Head of Center of Visualization and Satellite Information Technologies SRISA RAS (Moscow, Russian Federation)

Meyer B. – Dr.Sc., Professor, Head of Department in Swiss Federal Institute of Technology in Zurich, ETH
(Zurich, Switzerland)

Nguyen Thanh Nghi – Dr.Sc. (Physics and Mathematics), Professor, Vice-Principal of Hanoi Open University (Hanoi, Vietnam)

Nikolov R.V. – Full Professor of the University of Library Studies and Information Technology (Sofia, Bulgaria)

Osipov G.S. – Dr.Sc. (Physics and Mathematics), Professor, Deputy of the Principal of Institute of Systems Analysis
of the Russian Academy of Sciences (Moscow, Russian Federation)

Palyukh B.V. – Dr.Sc. (Engineering), Professor of Tver State Technical University (Tver, Russian Federation)

Rakhmanov A.A. – Dr.Sc. (Engineering), Professor, Deputy of the CEO of Concern "RTI Systems"
(Moscow, Russian Federation)

Serov V.S. – Dr.Sc. (Physics and Mathematics), Professor of the Oulu University of Applied Sciences (Oulu, Finland)

Sotnikov A.N. – Dr.Sc. (Physics and Mathematics), Professor, Joint Supercomputer Center of the Russian Academy
of Sciences (Moscow, Russian Federation)

Suleimanov D.Sh. – Academician of TAS, Dr.Sc. (Engineering), Professor of Kazan State Technical University
(Kazan, Republic of Tatarstan, Russian Federation)

Tarassov V.B. – Ph.D. (Engineering), Associate Professor of Bauman Moscow State Technical University
(Moscow, Russian Federation)

Taratoukhine V.V. – Ph.D. (Engineering), Dr.Ph., Managing Director of the Competence Centre ERP and ERCIS Lab
Russia of the ERCIS (Muenster, Germany)

Khoroshevsky V.F. – Dr.Sc. (Engineering), Professor of Moscow Institute of Physics and Technology
(Moscow, Russian Federation)

Yazenin A.V. – Dr.Sc. (Physics and Mathematics), Professor of Tver State University (Tver, Russian Federation)

ASSOCIATED EDITORIAL BOARD MEMBERS

Moscow Power Engineering Institute (Technical University), Moscow, Russian Federation

Technology Institute at Southern Federal University, Taganrog, Russian Federation

Tver State Technical University, Tver, Russian Federation

Research Institute CENTERPROGRAMSYSTEM, Tver, Russian Federation

EDITORIAL OFFICE ADDRESS

50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation

Phone: (482-2) 39-91-49 Fax: (482-2) 39-91-00

E-mail: red@cps.tver.ru

www.swsys.ru

Passed for printing 23.05.2017

Printed in printing-office "Faktor i K"

Lukina St. 4/1, Tver, 170028, Russian Federation

Published quarterly. 30th year of publication

Format 60×84 1/8. Circulation 1000 copies

Prod. order № 12. Wordage 180 pages. Price 257,40 rub.

Вниманию авторов

Международный журнал «Программные продукты и системы» публикует материалы научного и научно-практического характера по новым информационным технологиям, результаты академических и отраслевых исследований в области использования средств вычислительной техники. Практикуются выпуски тематических номеров по искусственному интеллекту, системам автоматизированного проектирования, по технологиям разработки программных средств и системам защиты, а также специализированные выпуски, посвященные научным исследованиям и разработкам отдельных вузов, НИИ, научных организаций.

Решением Президиума Высшей аттестационной комиссии (ВАК) Министерства образования и науки РФ международный журнал «Программные продукты и системы» внесен в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук.

Информация об опубликованных статьях по установленной форме регулярно предоставляется в систему Российского индекса научного цитирования (РИНЦ), в CrossRef и в другие базы и электронные библиотеки.

Условия публикации

К рассмотрению принимаются ранее нигде не опубликованные материалы, соответствующие тематике журнала (специализация 05.13.XX – Информатика, вычислительная техника и управление) и отвечающие редакционным требованиям.

Работа представляется в электронном виде в формате Word. При обилии сложных формул обязательно наличие статьи и в формате PDF. Формулы должны быть набраны в редакторе формул Word (Microsoft Equation или MathType). Объем статьи вместе с иллюстрациями – не менее 10 000 знаков. Диаграммы, схемы, графики должны быть доступными для редактирования (Word, Visio, Excel). Все иллюстрации для полиграфического воспроизведения представляются в черно-белом варианте. Цветные, тонированные, отсканированные, не подлежащие редактированию средствами Word рисунки и экранные формы следует присылать в хорошем качестве для их дополнительного размещения на сайте журнала в макете статьи с доступом по ссылке. (Публикация материалов с использованием гипертекста, графики, аудио-, видео-, программных средств и др. возможна в электронном издании «Программные продукты, системы и алгоритмы», сайт www.swsys-web.ru.) Заголовок должен быть информативным; сокращения, а также терминологию узкой тематики желательно в нем не использовать. Количество авторов на одну статью – не более 4, количество статей одного автора в номере, включая соавторство, – не более 2. Список литературы, наличие которого обязательно, должен включать не менее 10 пунктов.

Необходимы также содержательная структурированная аннотация (не менее 200 слов), ключевые слова (7–10) и индекс УДК. Название статьи, аннотация и ключевые слова должны быть переведены на английский язык (машинный перевод недопустим), а фамилии авторов, названия и юридические адреса организаций (если нет официального перевода), пристатейные списки литературы – транслитерированы по стандарту BGN/PCGN.

Вместе со статьей следует прислать сопроводительное письмо-рекомендацию в произвольной форме, экспертное заключение, лицензионное соглашение, а также сведения об авторах: фамилия, имя, отчество, название и юридический адрес организации, должность, ученые степень и звание (если есть), контактный телефон, электронный адрес, почтовый адрес для отправки бесплатного авторского экземпляра журнала.

Порядок рецензирования

Все статьи, поступающие в редакцию (соответствующие тематике и оформленные согласно требованиям к публикации), подлежат обязательному рецензированию в течение месяца с момента поступления.

В редакции есть устоявшийся коллектив рецензентов, среди которых члены международной редколлегии журнала, эксперты из числа крупных специалистов в области информатики и вычислительной техники ведущих вузов страны, а также ученые и специалисты НИИ «Центрпрограммсистем» (г. Тверь).

Рецензирование проводится конфиденциально. Автору статьи предоставляется возможность ознакомиться с текстом рецензии. При необходимости статья отправляется на доработку.

Рецензии обсуждаются на заседаниях рабочей группы, состоящей из членов научного совета журнала. Заседания проводятся раз в месяц в НИИ «Центрпрограммсистем» (г. Тверь), где принимается решение о целесообразности публикации статьи.

Статьи, одобренные редакционным советом, публикуются бесплатно в течение года с момента одобрения, а отправленные на доработку – с момента поступления после устранения замечаний.

Редакция международного журнала «Программные продукты и системы» в своей работе руководствуется сводом правил Кодекса этики научных публикаций, разработанным и утвержденным Комитетом по этике научных публикаций.

УДК 519.711.3

DOI: 10.15827/0236-235X.118.165-171

Дата подачи статьи: 15.02.17

2017. Т. 30. № 2. С. 165–171

ПОСТАНОВКА ЗАДАЧИ ФОРМИРОВАНИЯ НАПРАВЛЕНИЙ РАЗВИТИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ОРГАНИЗАЦИОННОГО ТИПА И АЛГОРИТМ ЕЕ РЕШЕНИЯ

В.А. Лясковский, д.т.н., профессор, советник генерального директора, niit@niit.tver.ru;

И.Б. Бреслер, к.в.н., доцент, генеральный директор, niit@niit.tver.ru;

М.А. Алашеев, к.т.н., специалист, niit@niit.tver.ru

*(Научно-исследовательский институт информационных технологий,
ул. Володарского, 3, г. Тверь, 170100, Россия)*

В статье рассматриваются постановка задачи формирования направлений развития автоматизированных систем обработки информации и управления организационного типа и алгоритм ее решения.

Необходимость решения данной задачи обусловлена тем, что многие автоматизированные системы создаются и эксплуатируются в течение десятилетий, при этом в процессе эксплуатации меняются требования, предъявляемые к этим системам, поэтому возникает потребность в периодическом формировании решений по приведению состояния автоматизированной системы в соответствие новым требованиям. В качестве основного показателя эффективности формируемых решений принят комплексный показатель, характеризующий степень автоматизации функциональных процессов, реализуемых в системе. В качестве ограничений выступают требования к обязательной автоматизации наиболее важных функциональных процессов и своевременности их выполнения, а также предельно допустимые финансовые и временные ресурсы процесса развития автоматизированной системы.

Проведенный анализ алгоритмической сложности решения задачи показывает невозможность ее решения путем рассмотрения всех возможных вариантов из-за экспоненциальной зависимости количества решений от размерности исходных данных. В связи с этим разработан эвристический алгоритм, позволяющий существенно сократить количество рассматриваемых вариантов и получить рациональное решение задачи при относительно небольшой вычислительной сложности.

Предложенный алгоритм позволяет обосновать решения по разработке и изготовлению комплексов средств автоматизации для оснащения органов управления из состава автоматизированных систем обработки информации и управления организационного типа, а также продления ресурса действующих в системе средств автоматизации. Предполагается реализовать алгоритм в автоматизированной системе поддержки принятия решений, представляющей собой программный комплекс, функционирующий на ПЭВМ потребительского класса.

Ключевые слова: автоматизированная система, проектирование систем управления, жизненный цикл систем управления, комплекс средств автоматизации.

Автоматизация процессов сбора, обработки, передачи информации и формирования управленческих решений в различных сферах человеческой деятельности, в том числе в деятельности силовых министерств и ведомств России, возможна с помощью автоматизированных систем обработки информации и управления организационного типа (АСОИУ ОТ). Они, как правило, представляют собой многоуровневые иерархические системы, состоящие из органов управления и каналов связи между ними.

Как показывает практика, АСОИУ ОТ создаются и эксплуатируются не одно десятилетие. Однако с течением времени меняются требования, предъявляемые к системам, возникают новые задачи и процессы, подлежащие автоматизации. Кроме того, устаревают и приходят в предельное состояние используемые комплексы средств автоматизации (КСА). Указанные факторы вызывают необходимость периодического решения задач по формированию управленческих решений, направленных на приведение АСОИУ ОТ в состояние, обеспечивающее выполнение актуальных требований.

Учитывая то, что АСОИУ ОТ состоят из большого количества разнотипных органов управления,

оснащенных различными средствами автоматизации, а также то, что процессы разработки, производства и поставки КСА реализуются в течение длительных промежутков времени в ходе выполнения соответствующих научно-исследовательских, опытно-конструкторских работ и серийных поставок и при этом все работы выполняются различными субъектами в условиях ресурсных ограничений, интуитивное решение подобной задачи не представляется возможным. В связи с этим возникает необходимость использования строгой методики формирования указанных решений и соответствующих этой методике программно-инструментальных средств, позволяющих получать требуемое решение в короткие сроки с минимальными трудозатратами.

Анализ имеющихся публикаций в рассматриваемой предметной области показывает, что вопросы управления развитием АСОИУ ОТ в настоящее время раскрыты не полностью. В ряде работ рассматриваются отдельные задачи формирования решений в части технического, программного и информационного обеспечения АСОИУ ОТ [1–6]. В других публикациях затрагиваются проблемы программно-целевого планирования НИОКР без учета выполнения функциональных требований к

АСОИУ ОТ [7, 8]. В связи с этим задача разработки подходов к формированию решений по развитию АСОИУ ОТ представляется актуальной и востребованной.

Суть выбора варианта решений по развитию АСОИУ ОТ [9–14] в том, что для каждого планового этапа прогнозирования должны быть выбраны системотехнические решения по автоматизации органов управления за счет оснащения (дооснащения) их серийными КСА (оснащение возможно в текущий момент времени) или разрабатываемыми КСА (оснащение возможно в будущем после окончания соответствующей опытно-конструкторской работы) с учетом допустимого (назначенного) ресурса использования КСА в составе АСОИУ ОТ. При этом при выборе варианта развития КСА должны быть также учтены требования по комплексной межуровневой автоматизации отдельных функциональных процессов, то есть выбор для реализации в органах управления таких функциональных задач, которые составляют рассматриваемый функциональный процесс.

С учетом изложенных требований в качестве целевой функции для решения рассматриваемой задачи целесообразно выбрать комплексный показатель [10, 11, 15–19], отражающий степень автоматизации требуемых функциональных процессов, при необходимости выполнения ряда из них в масштабе реального времени. Для этого в дальнейшем будем использовать два частных показателя:

- показатель оценки реализации функциональных процессов высшего приоритета;
- показатель оценки степени автоматизации остальных функциональных процессов.

Показатель оценки реализации функциональных процессов высшего приоритета (показатель 1) может быть записан как векторная дискретная булева функция, элементы которой принимают значение «1» при реализации функциональных процессов, «0» – в противном случае.

Показатель оценки степени автоматизации функциональных процессов (показатель 2) может быть записан как взвешенное по важности отношение реализованных процессов к сумме важностей функциональных процессов, требующих реализации.

При этом для показателей 1 и 2 под реализованным функциональным процессом будем понимать процесс, для которого реализованы все составляющие его функциональные задачи и выполняются требования к временным или вероятностно-временным характеристикам его выполнения (в случае, если такие требования для рассматриваемого процесса заданы).

В качестве исходных данных, используемых для решения рассматриваемой задачи, приняты следующие.

1. Организационная структура АСОИУ ОТ, описываемая как множество органов управления и

связей между ними. При этом каждый орган управления может быть оснащен не более чем одним КСА определенного типа.

2. Функциональная структура АСОИУ ОТ, описываемая как совокупность функциональных подсистем. Каждая из функциональных подсистем состоит из множества функциональных процессов, составными частями которых являются функциональные задачи.

В зависимости от предъявляемых требований к своевременности выполнения функциональные процессы подразделяются на функциональные процессы реального и нереального времени. Основное различие между указанными видами функциональных процессов состоит в том, что для функциональных процессов реального времени предъявляются требования к временным или вероятностно-временным характеристикам длительности их выполнения, а для функциональных процессов нереального времени такие требования не предъявляются.

В зависимости от важности, которая определяется степенью влияния реализуемого функционального процесса на общую эффективность применения АСОИУ ОТ по целевому назначению, все функциональные процессы подразделяются на функциональные процессы высшего приоритета и прочие функциональные процессы.

3. Типы органов управления, каждый из которых характеризуется множеством функциональных задач, подлежащих реализации в КСА.

4. Типы КСА, предназначенных для автоматизации органов управления. В зависимости от стадии жизненного цикла КСА разделяются на действующие (находящиеся на этапе эксплуатации в органах управления), серийные (находящиеся на этапе серийных поставок) и разрабатываемые (находящиеся на этапах выполнения соответствующих научно-исследовательских или опытно-конструкторских работ).

5. Текущий вариант оснащения органов управления КСА, в том числе включающий сведения о типах КСА, эксплуатируемых в органах управления, и их остаточном ресурсе.

6. Требования к временным и вероятностно-временным характеристикам выполнения функциональных процессов в АСОИУ ОТ.

7. Перечень, организационные, производственные и финансовые характеристики предприятий, которые выполняют работы или могут принимать участие в процессах разработки, производства и поставки КСА.

8. Перечень этапов прогнозирования с указанием их продолжительности и объемов финансирования.

Задача решается с учетом следующих условий и допущений:

- функциональный процесс в АСОИУ ОТ считается реализованным в том случае, когда реализо-

ваны все функциональные задачи, составляющие данный функциональный процесс, а также удовлетворены требования к временным и вероятностно-временным характеристикам его выполнения (если такие требования для рассматриваемого функционального процесса заданы);

- КСА для автоматизации органов управления представляются как неделимые элементы, в связи с чем выбор отдельных схмотехнических решений из состава КСА для оснащения органов управления не рассматривается;

- в рамках допустимого ресурса использования КСА всех типов, а также при принятии решения о продлении эксплуатации КСА надежные характеристики реализации функциональных процессов (функциональных задач) остаются неизменными и удовлетворяют заданным требованиям;

- помещения (сооружения) и объекты, в которых размещены (планируется разместить) средства автоматизации из состава АСОИУ ОТ, удовлетворяют требованиям по массогабаритным, климатическим и энергетическим показателям для размещения соответствующих типов КСА (если для ряда помещений и объектов требования по массогабаритным, климатическим и энергетическим показателям не выполняются, то к таким объектам предъявляются соответствующие требования, которые должны быть реализованы до момента постановки КСА на объекты автоматизации);

- каналы и сети передачи данных удовлетворяют информационным потребностям по обмену данными для реализации всех функциональных процессов и функциональных задач (если для ряда направлений обмена указанные информационные потребности не выполняются, то к элементам подсистемы передачи информации (объектам телекоммуникационной инфраструктуры) предъявляются соответствующие требования, которые должны быть реализованы до момента постановки КСА на объекты автоматизации);

- должностные лица органов управления имеют требуемый уровень квалификации (то есть при выполнении автоматизируемых функциональных задач и функциональных процессов время выполнения соответствующих функций управления должностными лицами органов управления не превышает заданное);

- объекты управления, которыми должна управлять АСОИУ ОТ, оснащены соответствующим окончательным оборудованием, обеспечивающим интеграцию КСА в контур управления без каких-либо доработок.

С учетом указанных исходных данных, допущений и ограничений формализованная постановка задачи формирования решений по развитию АСИОУ ОТ в условиях ресурсных ограничений может быть записана в следующем виде.

На каждом этапе прогнозирования необходимо определить вариант развития АСОИУ ОТ на ос-

нове выбора системотехнических решений $X^*(u)$, $Y^*(u)$, $Z^*(u)$, обеспечивающий максимизацию эффективности реализации функциональных процессов $\mathcal{E}(X(u), Y(u), Z(u))$, при обязательной реализации функциональных процессов высшего приоритета, при заданных предельно-допустимых временных и вероятностно-временных характеристиках реализации функциональных процессов, при выполнении ограничений на временные $T_{\text{доп}}$ и стоимостные $C_{\text{доп}}$ параметры процесса развития АСОИУ ОТ:

$$\langle X^*(u), Y^*(u), Z^*(u) \rangle = \underset{X(u), Y(u), Z(u)}{\text{Argmax}} \mathcal{E}(X(u), Y(u), Z(u))$$

при выполнении ограничений

$\Omega(X(u), Y(u)) \cap \Omega_b = \Omega_b$; $P_i(X(u), Y(u)) \geq P^{\text{пр}}_i(u)$; $\tau_i(X(u), Y(u)) \leq \tau^{\text{пр}}_i(u)$; $C(X(u), Y(u), Z(u)) \leq C_{\text{доп}}(u)$; $T(X(u), Y(u), Z(u)) \leq T_{\text{доп}}(u)$, где $u \in \{1, \dots, U\}$, U – количество плановых этапов прогнозирования развития АСОИУ ОТ; $X(u)$ – решения по разработке КСА; $Y(u)$ – решения по изготовлению КСА и оснащению органов управления; $Z(u)$ – решения по продлению эксплуатации КСА; $\Omega(X(u), Y(u))$ – множество реализованных функциональных процессов; Ω_b – множество функциональных процессов высшего приоритета; $P_i(X(u), Y(u))$ – вероятность своевременного выполнения i -го функционального процесса; $P^{\text{пр}}_i(u)$ – требуемая вероятность своевременного выполнения i -го функционального процесса; $\tau_i(X(u), Y(u))$ – среднее время выполнения i -го функционального процесса; $\tau^{\text{пр}}_i(u)$ – требуемое среднее время выполнения i -го функционального процесса; $C(X(u), Y(u), Z(u))$ – стоимость реализации решений по развитию АСОИУ ОТ [17]; $C_{\text{доп}}(u)$ – финансовые ограничения на развитие АСОИУ ОТ; $T(X(u), Y(u), Z(u))$ – продолжительность выполнения работ по развитию АСОИУ ОТ [20]; $T_{\text{доп}}(u)$ – длительность этапа прогнозирования.

Для анализа алгоритмической сложности задачи дополнительно введем обозначения: N – количество органов управления (объектов автоматизации) из состава АСОИУ ОТ; K_n – количество типов КСА для каждого органа управления, $n \in \{1, \dots, N\}$; M_n – количество функциональных задач каждого органа управления, $n \in \{1, \dots, N\}$.

Тогда общее количество вариантов разработки КСА, которое необходимо сформировать и оценить в процессе решения, будет рассчитываться в соответствии со следующим выражением:

$$Q_1 = \left(\prod_{n=1}^N (K_n + 1) 2^{M_n} \right)^U.$$

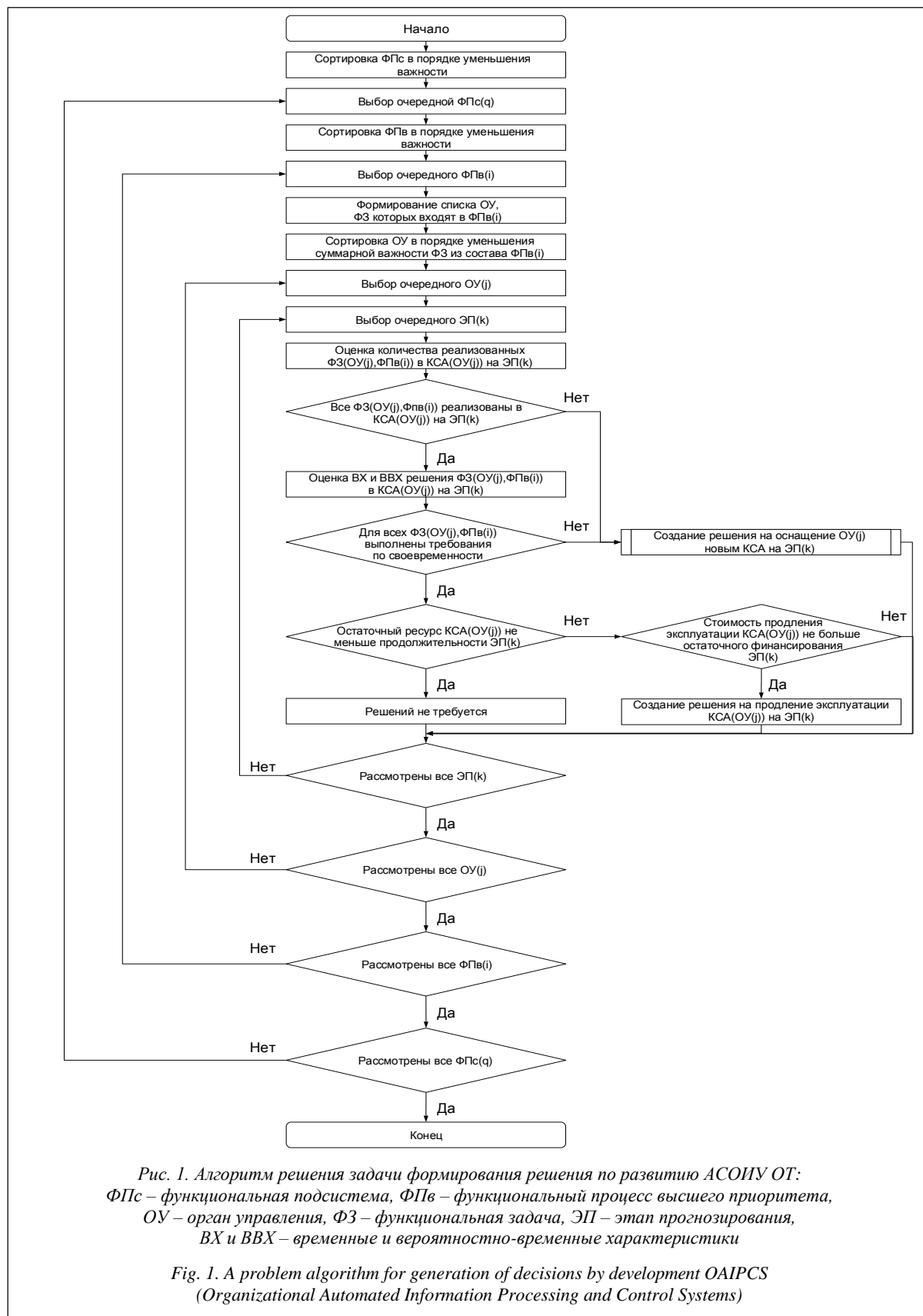
Общее количество вариантов изготовления КСА и оснащения органов управления будет рассчитываться в соответствии с формулой

$$Q_2 = \left(\prod_{n=1}^N (K_n + 1) \right)^U.$$

Общее количество вариантов продления эксплуатации КСА будет рассчитываться в соответствии с выражением $Q_3 = 2^{NU}$.

Общее количество вариантов решения задачи будет равно произведению трех вышеуказанных выражений: $Q = Q_1 \cdot Q_2 \cdot Q_3$.

Проведенный анализ алгоритмической сложности решения поставленной задачи показал, что данная задача является NP-трудной. При этом, как



показано в известных работах [5, 9, 10], даже с использованием высокопроизводительной вычислительной техники получить точное решение такой задачи возможно только для относительно небольших размерностей исходных данных. В связи с этим был предложен эвристический алгоритм решения (рис. 1), основанный на итеративно-последовательном формировании решения, начиная с реализации в АСОИУ ОТ наиболее важных функциональных процессов.

Сущность решения задачи с использованием предложенного алгоритма заключается в выполнении следующих действий:

1. Формирование базового варианта решения для каждого этапа прогнозирования.

1.1. Последовательное рассмотрение всех функциональных подсистем в порядке убывания их важности.

1.2. Последовательное рассмотрение всех функциональных процессов высшего приоритета в порядке убывания их важности для выбранной функциональной подсистемы.

1.3. Последовательное рассмотрение всех органов управления, функциональные задачи которых входят в выбранный функциональный процесс высшего приоритета, в порядке уменьшения суммарной важности функциональных задач.

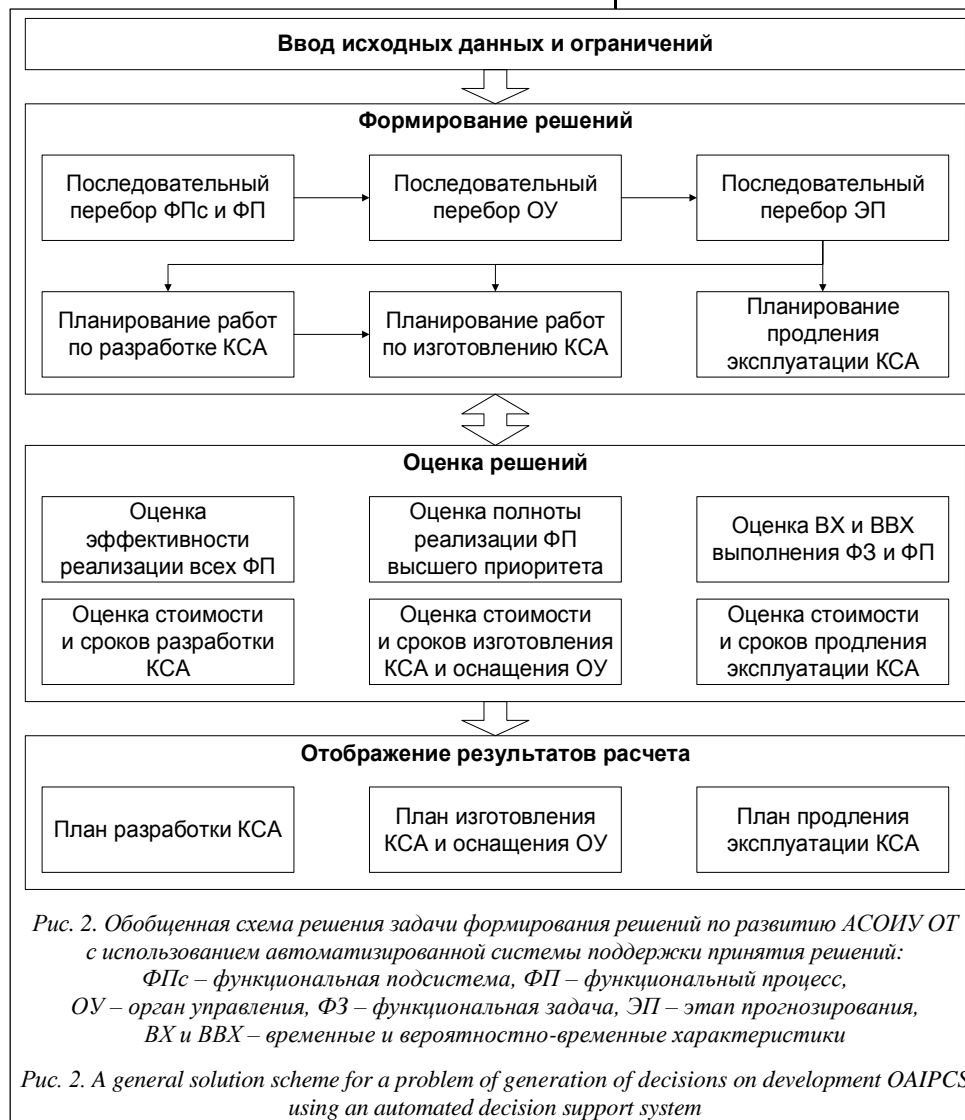
1.4. Для выбранного органа управления осуществляется последовательное рассмотрение всех этапов прогнозирования.

1.5. Для выбранного этапа прогнозирования осуществляется следующее.

1.5.1. Оценка полноты реализации всех функциональных задач выбранного функционального процесса высшего приоритета. Если реализованы не все функциональные задачи выбранного функционального процесса высшего приоритета, осуществляется переход к процедуре создания решения на оснащение объектов управления новым КСА на этапе прогнозирования, а затем переход к очередному этапу прогнозирования (п. 1.5). Процедура создания решения на оснащение органов управления новым КСА обеспечивает формирование планов работ по разработке и изготовлению КСА с

учетом выполнения ограничений на временные и стоимостные параметры процесса развития АСОИУ ОТ на каждом плановом этапе.

1.5.2. Оценка удовлетворения требований к своевременности решения (на основе оценки временных и вероятностно-временных характеристик) всех функциональных задач выбранного функционального процесса высшего приоритета в КСА. Если требования к своевременности выполняются не для всех функциональных задач выбранного функционального процесса высшего приоритета, то осуществляется переход к процедуре создания решения на оснащение органов управления новым КСА на этапе прог-



нозирования, а затем переход к очередному этапу прогнозирования (п. 1.5).

1.5.3. Оценка остаточного ресурса КСА на органах управления. Если остаточный ресурс КСА меньше продолжительности этапа прогнозирования и стоимость продления эксплуатации не превышает остаточный объем финансирования этапа прогнозирования, создается решение на продление эксплуатации КСА. Затем осуществляется переход к очередному этапу прогнозирования (п. 1.5).

1.6. Если рассмотрены все этапы прогнозирования, осуществляется переход к очередному органу управления (п. 1.4).

1.7. Если рассмотрены все органы управления, осуществляется переход к очередному функциональному процессу высшего приоритета (п. 1.3).

1.8. Если рассмотрены все функциональные процессы высшего приоритета, осуществляется переход к очередной функциональной подсистеме (п. 1.2).

1.9. Если рассмотрены все функциональные подсистемы, базовый вариант решения сформирован и осуществляется переход к п. 2.

2. Формирование рационального варианта для каждого этапа прогнозирования.

Выполняются действия по п. 1.1–1.9, при этом все функциональные процессы высшего приоритета считаются реализованными, и последовательно (в соответствии со значением параметра важности) рассматриваются функциональные процессы не из состава функционального процесса высшего приоритета, задача решается в условиях оставшихся ресурсов.

Данный алгоритм предполагается реализовать в разрабатываемой автоматизированной системе поддержки принятия решений, что позволит упростить процедуры задания исходных данных, проведения расчетов и отображения результатов решения. Обобщенная схема решения поставленной задачи с использованием предлагаемой автоматизированной системы поддержки принятия решений представлена на рисунке 2.

Для программной реализации предлагаемой автоматизированной системы поддержки принятия решений выбран следующий технологический стек: операционная система Astra Linux Special Edition 1.4, библиотеки программных компонентов Qt 4.8.6, система управления БД PostgreSQL 9.3.

Таким образом, в настоящей статье рассмотрены постановка задачи формирования направлений развития АСОИУ ОТ и алгоритм ее решения, применение которого позволит обосновать решения по разработке и изготовлению КСА для оснащения органов управления из состава АСОИУ ОТ, а также продления ресурса действующих в системе КСА.

Литература

1. Бушуев С.Н., Осадчий А.С., Фролов В.М. Теоретические основы создания информационно-технических систем. СПб: ВАС, 1998. 382 с.
2. Шпак В.Ф., Директоров Н.Ф., Мирошников В.И. [и др.]. Информационные технологии в системе управления ВМФ. СПб: Элмор, 2005. 832 с.
3. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. М.: СИНТЕГ, 2002. 268 с.
4. Бородакий Ю.В., Боговик А.В., Курносое В.И. и др. Основы теории управления в системах специального назначения. М., 2008. 400 с.
5. Ляковский В.Л. Системотехнические основы автоматизации процессов обработки информации и управления в иерархических системах военного назначения. Тверь: Изд-во ВА ВКО, 2014. 244 с.
6. Khaled M. Khan, Yan Zhang. Managing Corporate information systems evolution and maintenance. IGI Publ., 2005, 376 p.
7. Буренок М.В., Ляпунов В.М., Мудров В.И. Теория и практика планирования и управления развитием вооружения: монография. М.: Вооружение, политика, конверсия, 2004. 418 с.
8. Буренок В.М., Ивлеев А.А., Корчак В.Ю. Программно-целевое планирование и управление созданием научно-технического задела для перспективного и нетрадиционного вооружения: монография. М.: Граница, 2007. 408 с.
9. Пильщиков Д.Е. Методы и методики создания перспективных КСА для пунктов (органов) управления ВВС на основе применения типовых проектных решений и процедур. Тверь: Изд-во ВА ВКО, 2005. 200 с.
10. Ляковский В.Л., Алашеев М.А., Пильщиков Д.Е. и др. Методика выбора состава задач и комплексов средств автоматизации для многоуровневой системы управления РЭС // Радиотехника. 2004. № 10. С. 75–78.
11. Ляковский В.Л. Методологические основы создания (развития) и оснащения иерархических автоматизированных систем специального назначения. Тверь: Изд-во ВА ВКО, 2010. 284 с.
12. Ляковский В.Л., Алашеев М.А., Морозов О.Г., Потапов В.Н. Вопросы создания интегрированных автоматизированных систем организационного управления // Инфокоммуникационные технологии. 2007. № 2. С. 62–64.
13. Лепешкин О.М., Корсунский А.С. Оптимизация структуры комплекса информационно-технических средств в автоматизированных системах управления // Автоматизация процессов управления. 2011. № 4. С. 76–81.
14. Арепин Ю.И., Допира Р.В., Смоляков А.А. Военная кибернетика: методология создания автоматизированных систем управления техническим обеспечением. Тверь: Изд-во НИИ ЦПС, 2006. 203 с.
15. Зальмарсон А.Ф., Юдин А.Ж. Комплексный подход к оценке эффективности программно-аппаратного комплекса автоматизации деятельности органов военного управления ВМФ // Автоматизация процессов управления. 2015. № 3. С. 4–11.
16. Митрушкин Е.И. Комплекс приближенных оценок характеристик системы массового обслуживания // Научн. вестн. МИРЭА. 2013. № 1 (13). С. 59–65.
17. Костокрызов А.И., Липаев В.В. Сертификация качества функционирования автоматизированных информационных систем. М.: Вооружение. Политика. Конверсия, 1996. 279 с.
18. Krogstie J. Model-based development and evolution of information systems: a quality approach. Springer Science & Business Media, 2012, 442 p.
19. Gorla N., Somers T.M., Wong B. Organizational impact of system quality, information quality, and service quality. Jour. Strategic Information Systems, 2010, no. 19, pp. 207–228.
20. Липаев В.В. Технико-экономическое обоснование проектов сложных программных средств. М.: СИНТЕГ, 2004. 284 с.

STATEMENT OF THE PROBLEM OF FORMATION OF DIRECTIONS OF DEVELOPMENT
OF ORGANIZATIONAL AUTOMATED SYSTEMS AND ITS SOLUTION ALGORITHM

V.L. Lyaskovsky¹, Dr.Sc. (Engineering), Professor, niiit@niiit.tver.ru

I.B. Bresler¹, Ph.D. (Military Science), Associate Professor, niiit@niiit.tver.ru

M.A. Alasheev¹, Ph.D. (Engineering), niiit@niiit.tver.ru

¹ Research Institute of Information Technologies, Volodarskogo St. 4, Tver, 170100, Russian Federation

Abstract. The article discusses the statement of the problem of formation of directions for the development of organizational automated information processing and control systems and its solution algorithm.

It is necessary to solve this problem due to the fact that many automated systems are created and operated for decades, while the requirements for these systems change over time. Therefore, there is a need to form solutions in order to bring an automated system in compliance with new requirements from time to time. The main performance indicator of generated solutions is a complex index of the degree of automation of functional processes implemented in the system. The constraints are mandatory requirements for automation of the most important functional processes and timeliness of their implementation, as well as the maximum permissible financial and time resources for development of an automated system.

The analysis of the algorithmic complexity of the problem solution shows the impossibility of its solution by considering all possible options due to exponential dependence of the number of decisions on the dimension on the source data. Therefore, the authors have developed a heuristic algorithm to reduce the number of options under consideration and rational solution to obtain a relatively small computational complexity.

The proposed algorithm allows getting the decision on the development and production of complex automation equipment for an automated control system, as well as extending the life of existing automation. This algorithm is expected to be implemented in an automated decision support system, which functions as software on a consumer-grade personal computer.

Keywords: automated system, automated control system design, life cycle of automated control systems, complex automation equipment.

References

1. Bushuev S.N., Osadchy A.S., Frolov V.M. *Teoreticheskie osnovy sozdaniya informatsionno-tehnicheskikh sistem* [Theoretical Basics of Information Technology System Creating]. St. Petersburg, VAS Publ., 1998, 382 p.
2. Shpak V.F., Direktorov N.F., Miroshnikov V.I. *Informatsionnye tekhnologii v sisteme upravleniya VMF* [Information Technologies in the Navy Control System]. St. Petersburg, Elmor Publ., 2005, 832 p.
3. Lipaev V.V. *Sistemnoe projektirovanie slozhnykh programmykh sredstv dlya informatsionnykh sistem* [System Design of Complex Software for Information Systems]. Moscow, SINTEG Publ., 2002, 268 p.
4. Borodaky Yu.V., Bogovik A.V., Kurnosov V.I. *Osnovy teorii upravleniya v sistemakh spetsialnogo naznacheniya* [Fundamentals of the Control Theory in Special-Purpose Systems]. Moscow, 2008, 400 p.
5. Lyaskovsky V.L. *Sistemotekhnicheskie osnovy avtomatizatsii protsessov obrabotki informatsii i upravleniya v ierarkhicheskikh sistemakh voennogo naznacheniya* [System Basics of Automation of Information Processing and Management in Hierarchical Systems for Military Use]. Tver, VA VKO Publ., 2014, 244 p.
6. Khaled M. Khan, Yan Zhang. *Managing Corporate Information Systems Evolution and Maintenance*. IGI Publ., 2005, 376 p.
7. Burenok M.V., Lyapunov V.M., Mudrov V.I. *Teoriya i praktika planirovaniya i upravleniya razvitiem vooruzheniya* [Theory and Practice of Planning and Management of Weapons Development]. Moscow, Vooruzhenie, politika, konversiya Publ., 2004, 418 p.
8. Burenok V.M., Ivlev A.A., Korchak V.Yu. *Programno-tselevoe planirovanie i upravlenie sozdaniem nauchno-tehnicheskogo zadela dlya perspektivnogo i netraditsionnogo vooruzheniya* [Program-Oriented Planning and Management of Creating a Scientific and Technical Basis for Long-Term and Non-Conventional Weapons]. Moscow, Granitsa Publ., 2007, 408 p.
9. Pilshchikov D.E. *Metody i metodiki sozdaniya perspektivnykh KSA dlya punktov (organov) upravleniya VVS na osnove primeneniya tipovykh proyektnykh resheniy i protsedur* [Methods and Techniques for Creating Advanced Air Force Complex Automated Control Systems on the Basis of Standard Design Solutions and Procedures]. Tver, 2005, 200 p.
10. Lyaskovsky V.L., Alasheev M.A., Pilshchikov D.E. Method of selection of Φ composition of tasks and a complex of automation means for a multi-level radio-electronic control system. *Radiotekhnika* [Radio Engineering]. 2004, no. 10, pp. 75–78 (in Russ.).
11. Lyaskovsky V.L. *Metodologicheskie osnovy sozdaniya (razvitiya) i osnashcheniya ierarkhicheskikh avtomatizirovannykh sistem spetsialnogo naznacheniya* [Methodological Basics of Creation (Development) and Equipment of the Hierarchical Special Purpose Automated Systems]. Tver, 2010, 284 p.
12. Lyaskovsky V.L., Alasheev M.A., Morozov O.G., Potapov V.N. Issues of development organizational management integrated automated systems. *Infokommunikatsionnye tekhnologii* [Infocommunication technologies]. 2007, no. 2, pp. 62–64 (in Russ.).
13. Lepeshkin O.M., Korsunsky A.S. Structure optimization for software and hardware facilities in computer-aided control systems. *Avtomatizatsiya protsessov upravleniya* [Automation of Control Processes]. 2011, no. 4 (26), pp. 76–81 (in Russ.).
14. Arepin Yu.I., Dopira R.V., Smolyakov A.A. *Voennaya kibernetika: metodologiya sozdaniya avtomatizirovannykh sistem upravleniya tekhnicheskimi obespecheniyami* [Military Cybernetics: Methodology of Creation of Automated Technical Support Control Systems]. ZAO NII CPS Publ., Tver, 2006, 203 p.
15. Zalmanson A.F., Yudin A.G. A comprehensive approach to the assessment of the effectiveness of software-hardware complex of the naval authorities functional activity automation. *Avtomatizatsiya protsessov upravleniya* [Automation of Control Processes]. 2015, no. 3 (41), pp. 4–11 (in Russ.).
16. Mitrushkin E.I. A set of approximate estimates of queuing system characteristics. *Nauchn. vestn. MIREA* [Scientific Herald of MIREA]. Moscow, 2013, no. 1 (13), pp. 59–65 (in Russ.).
17. Kostogryzov A.I. *Sertifikatsiya kachestva funktsionirovaniya avtomatizirovannykh informatsionnykh sistem* [Quality Certification of Automated Information Systems Functioning]. Moscow, Vooruzhenie, politika, konversiya Publ., 1996, 275 p.
18. Krogstie J. *Model-Based Development and Evolution of Information Systems: A Quality Approach*. Springer Science & Business Media Publ., 2012, 442 p.
19. Gorla N., Somers T.M., Wong B. Organizational impact of system quality, information quality, and service quality. *Jour. of Strategic Information Systems*. 2010, no. 19 (3), pp. 207–228.
20. Lipaev V.V. *Tekhniko-ekonomicheskoe obosnovanie projektov slozhnykh programmykh sredstv* [Feasibility Study of Complex Software Projects]. Moscow, SINTEG Publ., 2004, 284 p.

УДК 519.87+621.311

DOI: 10.15827/0236-235X.118.2.172-179

Дата подачи статьи: 13.02.17

2017. Т. 30. № 2. С. 172–179

ФОРМИРОВАНИЕ ВАРИАНТОВ РАЗВИТИЯ ЭНЕРГЕТИКИ ВЬЕТНАМА МЕТОДАМИ КОМБИНАТОРНОГО МОДЕЛИРОВАНИЯ

А.В. Еделева, к.т.н., старший научный сотрудник, flower@isem.irk.ru;

В.И. Зоркальцев, д.т.н., главный научный сотрудник, zork@isem.irk.ru

*(Институт систем энергетики им. Л.А. Мелентьева СО РАН,
ул. Лермонтова, 130, г. Иркутск, 664033, Россия);*

Доан Ван Бинь, к.т.н., директор института, doanbinh@ies.vast.vn;

Нгуен Хоай Нам, зав. лабораторией, nhnam@ies.vast.vn

(Институт энергетической науки ВАНТ, ул. Хоанг Куок Вьет, А9, 18, г. Ханой, Вьетнам)

В статье рассмотрен подход комбинаторного моделирования к исследованию вариантов развития энергетики страны. Основу подхода составляет представление развития моделируемой системы в форме направленного графа, узлы которого соответствуют возможным состояниям системы в отдельные моменты времени, а связи характеризуют допустимость переходов из одного состояния в другое.

Методы комбинаторного моделирования, являясь наглядной формой представления динамических дискретных ветвящихся альтернатив, позволяют имитировать долгосрочный процесс развития исследуемой системы при различных возможных внешних и внутренних условиях, учитывая дискретность будущих состояний системы, определять рациональную стратегию развития исследуемой системы. Процедуры формирования и анализа графа развития энергетики страны реализованы в виде пакета прикладных программ «Корректива». Анализ графов развития больших размеров проводится с помощью распределенных вычислений.

Приводится описание исследования устойчивого развития энергетики Вьетнама и его энергетической безопасности с 2015 по 2030 гг., выполненного в 2015 г. Институтом энергетической науки Вьетнамской академии науки и технологий. На основе данных этого исследования показано применение методов комбинаторного моделирования для формирования и анализа вариантов устойчивого развития энергетики Вьетнама с учетом требований энергетической безопасности. Создан граф развития топливно-энергетического комплекса Вьетнама, состоящий из 531 442 узлов. Его обсчет проводился в распределенной вычислительной среде, состоящей из вычислительных узлов высокопроизводительного кластера Института динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук.

На основе результатов расчета была определена оптимальная стратегия развития энергетики Вьетнама, обеспечивающая минимум затрат на развитие и функционирование топливно-энергетического комплекса в рассматриваемый период времени.

Ключевые слова: комбинаторное моделирование, система энергетики, топливно-энергетический комплекс, поддержка принятия решений, распределенная вычислительная среда.

Разработку долгосрочных программ развития энергетики с учетом фактора неопределенности (неоднозначности) исходной информации и условий развития [1] необходимо проводить на основе общенергетических исследований [2, 3] с применением специальных методов, моделей, БД, вычислительных средств. В моделях должен рассматриваться длительный период времени (до 30–40 лет) с выделением нескольких этапов развития и функционирования систем энергетики. Также в моделях в явном виде должна учитываться дискретность вариантов развития объектов и систем энергетики. Используемые для формирования вариантов развития энергетики и последующего их анализа инструменты должны быть хорошо обоснованными (то есть давать доказательные утверждения) и гибкими. Их основу должны составлять некоторые общие организующие исследования, алгоритмы формирования и отбора вариантов. При этом следует допускать возможность применения разных расчетных схем и моделей для исследования конкретных проблем.

Эффективным инструментом для анализа отдельных вариантов развития энергетики стала ба-

лансовая экономико-математическая модель *топливно-энергетического комплекса* (ТЭК) [4]. По принципам построения и приемам описания связей и свойств систем энергетики данная модель похожа на обобщенную сетевую модель систем энергетики США [5], а по своим возможностям она не уступает ни модели развития электроэнергетики и теплоснабжения в рамках ТЭК страны EPOS [6] (часть модельно-информационного комплекса SCANNER [7]), ни зарубежным моделям MARKAL [8], MESSAGE [9], EFOM [10], TIMES [11], Balmorel [12].

Данная модель ТЭК позволяет рассматривать энергетику страны как единое целое от производства энергетических ресурсов до потребления конечных энергоносителей в основных отраслях экономики, включая все стадии переработки и преобразования энергоресурсов, а также исследовать процесс развития технологической и территориальной структуры энергетики.

Модель ТЭК включает в себя модели основных систем энергетики: газоснабжения, углеснабжения, нефте- и нефтепродуктоснабжения, электро-энергетики. Наряду с производственным и распре-

делительным (транспортным) блоками присутствует блок потребления, в котором представлены основные потребители продукции отраслей ТЭК, ранжированные по соответствующим категориям.

В ходе разработки методики поиска рациональных путей развития ТЭК с учетом требований *энергетической безопасности* (ЭБ) стало очевидно, что в рамках одной модели ТЭК невозможно рассмотреть и описать все характерные сочетания внешних условий с вариантами развития и функционирования ТЭК с учетом неопределенности, воздействия угроз ЭБ и прочих факторов. Это приводит к большому числу расчетных состояний ТЭК, что может создавать существенные трудности при формировании и анализе классическими методами исследования. Для решения этой проблемы были использованы методы комбинаторного моделирования, которые, являясь наглядной формой представления динамических дискретных ветвящихся альтернатив, позволяют имитировать долгосрочный процесс развития исследуемой системы при различных возможных внешних и внутренних условиях, учитывая дискретность будущих состояний системы, определять рациональную стратегию развития исследуемой системы.

В данной статье описывается программное обеспечение, реализующее некоторые процедуры комбинаторного моделирования, а в качестве примера рассмотрено их применение для исследования проблем устойчивого развития ТЭК Вьетнама.

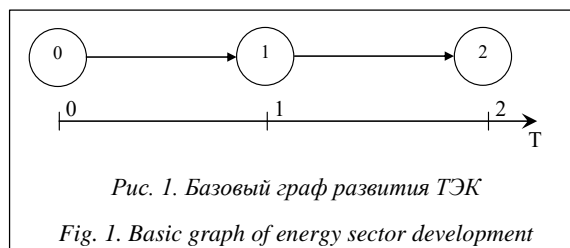
Формирование вариантов развития ТЭК методами комбинаторного моделирования

Основу комбинаторного моделирования составляет представление развития каждого объекта моделируемой системы в форме направленного графа, узлы которого соответствуют возможным состояниям объекта в отдельные моменты времени, а связи характеризуют допустимость переходов из одного состояния в другое. Формирование и анализ вариантов развития системы в целом проводится путем рассмотрения различных сочетаний состояний и переходов отдельных объектов, поэтому такой подход был назван комбинаторным моделированием [13].

Под объектами понимаются структурные единицы моделируемой системы, рассматриваемые как единое целое. Это могут быть предприятия или совокупности однотипных предприятий отраслей ТЭК, предприятия-потребители энергоресурсов или обобщенно некоторые наборы (категории) потребителей. Степень агрегирования в выделении объектов предопределяется постановкой решаемой проблемы и возможностями информационного обеспечения. Причем в рамках рассматриваемого подхода можно естественным образом проводить объединение объектов. Совокупность нескольких

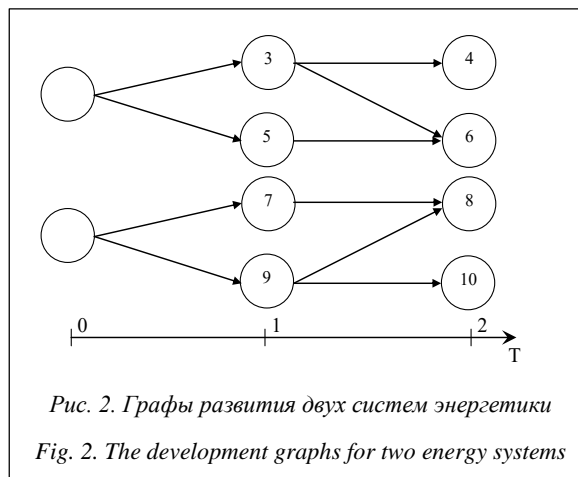
объектов в одной задаче может образовать единый объект при рассмотрении более крупных проблем.

На первом этапе исследования создается так называемый базовый граф развития ТЭК, содержащий по одному узлу для каждого опорного года (интервала времени T). С каждым узлом связывается основная информация для создания возможных состояний ТЭК со своими особенностями топливо- и энергоснабжения потребителей. На рисунке 1 представлен базовый граф для трех моментов времени, где в центре кругов показаны номера узлов. Узел с номером 0 описывает начальное состояние ТЭК.



На втором этапе структура ТЭК разбивается на несколько составляющих частей (объектов), например, по территориальному или отраслевому признаку. Для каждого объекта экспертами строится свой граф изменений их параметров по опорным годам, который должен иметь один исходный узел, описывающий начальное состояние объекта.

На рисунке 2 показаны графы развития двух систем энергетики. Исходные узлы, соответствующие моменту времени 0, не имеют номера, так как в последующей генерации графа развития ТЭК участия не принимают.



На третьем этапе строится граф развития ТЭК путем комбинирования информации об изменениях параметров различных частей ТЭК, принадлежащих одному временному срезу, и наложения полученного набора изменений на базовую информацию, связанную с соответствующим узлом базового графа. Таким образом, получается множество возможных состояний ТЭК для определенного мо-

мента времени, полученные состояния ТЭК связываются между собой дугами-переходами.

На основе комбинирования узлов и дуг графов (рис. 1 и 2) создается граф развития ТЭК, представленный на рисунке 3, где над его узлами показаны сочетания комбинируемых номеров узлов. В центре узлов указаны номера сгенерированных возможных состояний ТЭК (узлов графа развития ТЭК).

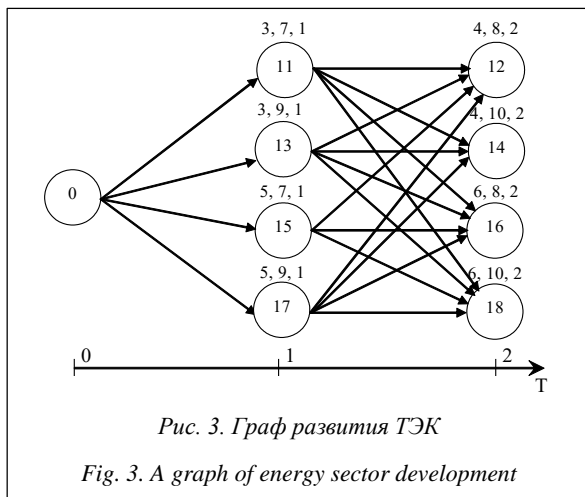


Рис. 3. Граф развития ТЭК

Fig. 3. A graph of energy sector development

Все пути развития ТЭК начинаются из единого начального состояния ТЭК, которое является исходным узлом базового графа (рис. 1).

На четвертом этапе от исходного узла будет осуществляться проход по графу для оценки допустимости состояний и переходов в графе развития ТЭК, так как не всякие возможные состояния и переходы в графе могут быть допустимыми. Для этой цели в комбинаторном моделировании присутствуют общесистемные ограничения, среди которых можно выделить два типа:

- логические условия; некоторые варианты в развитии одного объекта могут осуществляться только при реализации определенных вариантов в развитии других объектов;
- балансовые и другие расчетные ограничения; это ограничения на объемы использования лимитированных ресурсов и производимой продукции для каждого момента времени и каждого перехода; такие ограничения могут быть заданы в виде балансовых уравнений или неравенств.

Для реализации логических условий используются списки пар несовместных узлов. Парой несовместных узлов называется пара узлов, принадлежащих графам развития различных объектов моделируемой системы, сочетание которых в возможном состоянии системы по каким-либо причинам невозможно или не имеет смысла.

Представленная выше балансовая экономико-математическая модель ТЭК относится ко второму типу общесистемных ограничений. Допустимость возможного состояния ТЭК напрямую зависит от корректности результатов расчета на ней.

Уровень ЭБ в любом возможном состоянии ТЭК для выполнения требований ЭБ может оцениваться с использованием аппарата индикаторов. Значения индикаторов ЭБ вычисляются на основе результатов расчета на модели ТЭК. Уровень ЭБ определяется путем сравнения значений индикаторов и их пороговых значений.

В графе развития ТЭК, представленном на рисунке 3, проверку допустимости по разным причинам не прошли четыре узла (рис. 4).

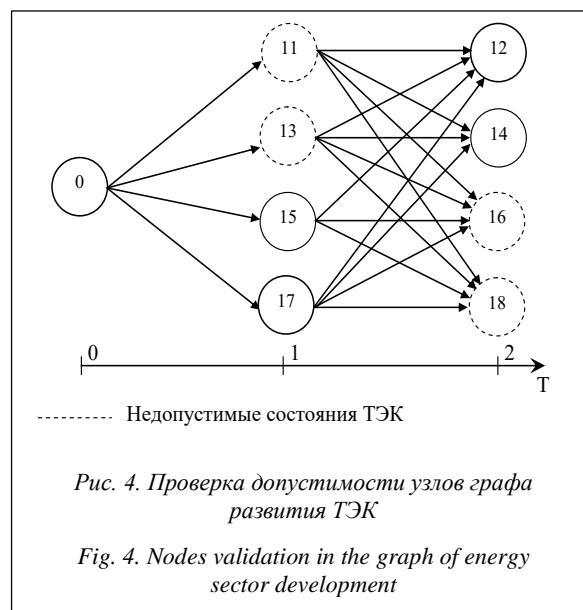


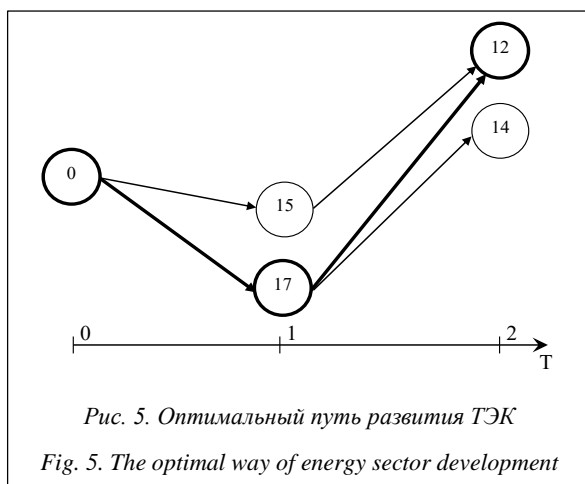
Рис. 4. Проверка допустимости узлов графа развития ТЭК

Fig. 4. Nodes validation in the graph of energy sector development

На пятом этапе строится граф развития ТЭК, содержащий допустимые состояния и переходы. Сначала проходом от исходного до конечных узлов определяются состояния и переходы, недостижимые из начального состояния. Затем обратным проходом определяются тупиковые состояния и переходы, по которым невозможно построить пути развития ТЭК от начального до последнего момента интервала времени T . Недопустимые, недостижимые и тупиковые состояния и переходы удаляются из графа развития ТЭК.

На последнем этапе в графе развития ТЭК по какому-либо критерию путем поиска кратчайших путей из исходного узла до конечных формируется множество оптимальных и близких к ним путей развития ТЭК. На рисунке 5 показан граф развития ТЭК, содержащий допустимые состояния и переходы и сформированный на основе графа, представленного на рисунке 4. На нем узлы и дуги оптимального пути, обеспечивающего минимум затрат на развитие и функционирование ТЭК, выделены жирными линиями.

Основная проблема алгоритмов комбинаторного моделирования при их реализации заключается в необходимости рассмотрения большого числа состояний и переходов в развитии моделируемой системы. Число это возрастает экспоненциально с ростом числа выделяемых объектов и их



состояний. Применение этого подхода возможно только при наличии возможностей по скоростной обработке больших объемов данных, что в настоящее время обеспечивается организацией распределенных вычислений, а также за счет использования экономных алгоритмов расчета. Существенное значение имеют разумное агрегирование объектов и состояний моделируемой системы на этапе задания исходной информации, а также организация процесса выделения допустимых вариантов.

Пакет прикладных программ «Корректива»

Вышеописанные процедуры формирования и анализа вариантов развития ТЭК реализованы в

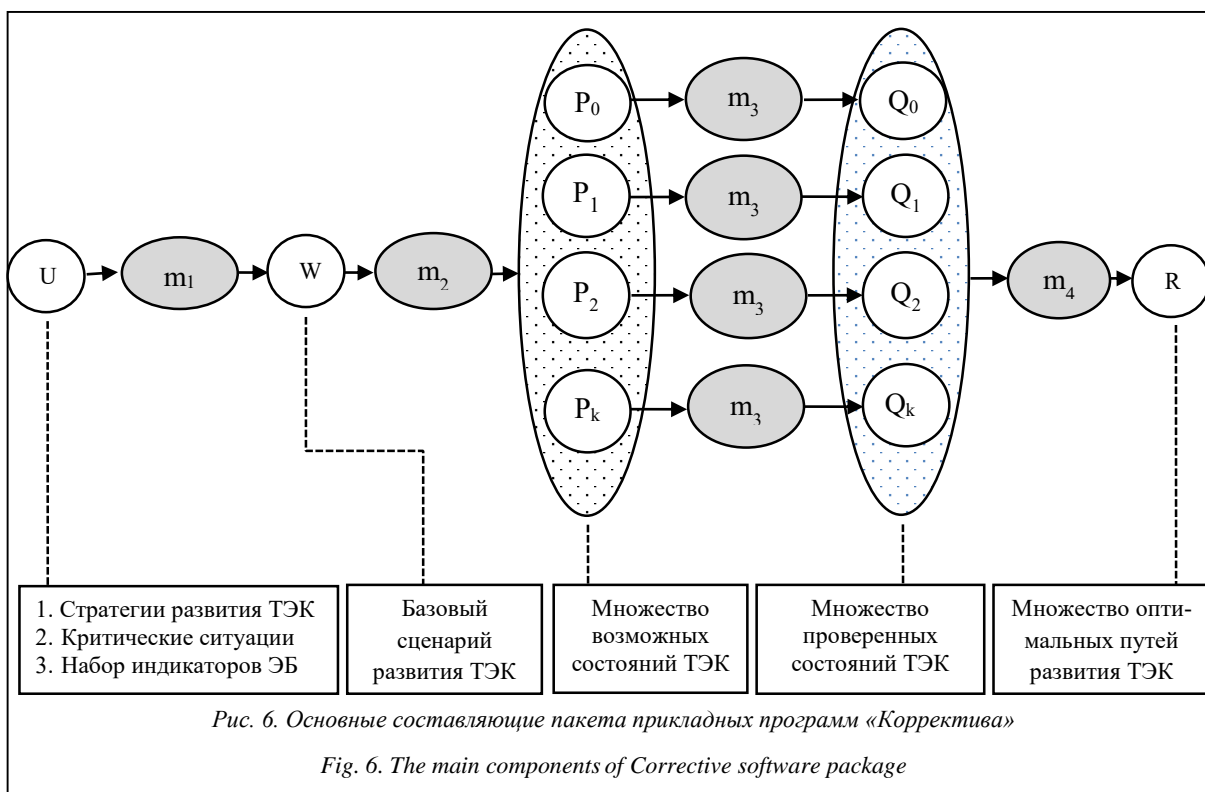
виде пакета прикладных программ «Корректива» [14], в котором можно выделить следующие основные программные модули:

- модуль формирования базовых сценариев развития ТЭК (m_1);
- модуль создания графа развития ТЭК (m_2);
- модуль оценки допустимости состояния ТЭК (m_3);
- модуль поддержки экспертного анализа множества оптимальных и близких к ним путей развития ТЭК (m_4).

Информационно-логические связи объектов предметной области пакета представлены на рисунке 6 в виде двудольного ориентированного графа. Модули m_1 , m_2 , m_3 , m_4 представлены серыми овалами, а входные и выходные параметры – окружностями.

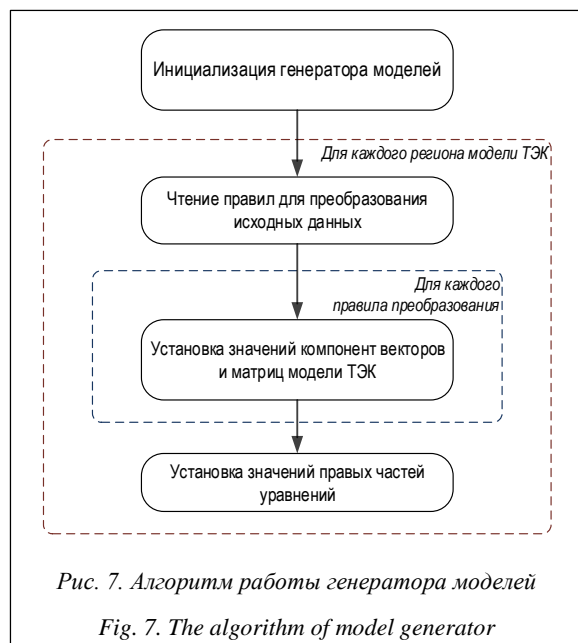
Основной задачей модуля m_1 является чтение информации U из БД и формирование на ее основе согласно целям исследования одного из конкретных сценариев развития ТЭК W . Блок исходной информации U содержит множество исследуемых стратегий развития ТЭК и может включать перечень критических ситуаций (сценариев возмущений), которые могут возникнуть в течение периода времени T вследствие реализации угроз ЭБ, и набор индикаторов ЭБ.

Модуль m_2 реализует методы комбинаторного моделирования, отвечающие за формирование графа развития ТЭК и его обход. Сценарий развития ТЭК W модулем m_2 преобразуется в базовый граф развития ТЭК.



Стратегии развития ТЭК, сценарии возмущений реализуются изменением по определенным законам значений компонент матриц и векторов, составляющих систему уравнений модели ТЭК. Подобным образом модуль m_2 на основе базовых состояний ТЭК создает выходное множество P , состоящее из возможных состояний ТЭК: P_0, P_1, P_2, P_K .

Ядро модуля m_2 представляет собой несколько десятков скриптов на интерпретируемом языке программирования Lua [15]. Ключевым компонентом m_2 является генератор моделей [16], который создает новое состояние ТЭК. Генератор моделей (рис. 7) управляется набором правил, преобразующих данные сценария развития ТЭК W в векторы и матрицу A уравнений. Исследователь имеет возможность изменять правила преобразования исходной информации.



Модуль m_3 проводит оценку допустимости состояния ТЭК из множества P с помощью многоступенчатой системы ограничений. На первом этапе происходит проверка логических условий. На втором этапе осуществляется решение модели ТЭК как задачи линейного программирования. Если решение существует, то далее в случае необходимости на основе результатов решения вычисляются значения индикаторов ЭБ и осуществляется проверка уровня ЭБ.

Результаты работы системы ограничений модуля m_3 , добавляясь к P_0, P_1, P_2, P_K , образуют выходное множество Q , состоящее из проверенных состояний ТЭК Q_0, Q_1, Q_2, Q_K .

Модуль m_4 на основе полученного множества Q_0, Q_1, Q_2, Q_K формирует граф развития ТЭК, содержащий допустимые состояния и переходы, а также множество оптимальных и близких к ним пу-

тей развития ТЭК, с которым далее работает исследователь с целью проведения детального анализа.

После создания графа развития ТЭК каждый его узел является полностью независимым от других с точки зрения проведения расчетов, что обеспечивает возможность организации на данном этапе многовариантных расчетов с параллелизмом по данным [17]. Для организации процесса распределенных вычислений используется инструментальный комплекс DISCOMP [18], ориентированный на автоматизацию разработки и применения распределенных пакетов прикладных программ в разнородных распределенных вычислительных средах.

Моделирование устойчивого развития ТЭК Вьетнама

На базе описанной выше модели ТЭК в период с 2011 по 2015 гг. в ходе совместных исследований, проведенных Институтом систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук и Институтом энергетической науки Вьетнамской академии науки и технологий, была разработана модель развития ТЭК Вьетнама [19]. В ней структура ТЭК представлена по восьми агрегированным регионам: дельта Красной реки, Северо-Восточный, Северо-Западный, север Центрального побережья, юг Центрального побережья, Центральное нагорье, Юго-Восточный и дельта Меконга. Входные данные модели включают в себя технико-экономические показатели энергоснабжения (затраты, стоимость и объемы производства, импорта и экспорта), преобразования и транспорта энергоресурсов, потребления энергоресурсов по видам, включая уголь, нефть, газ и электроэнергию.

Институтом энергетической науки с помощью специализированного модуля m_1 программного пакета «Корректив» [20] в 2015 г. были проведены исследования устойчивого развития энергетики Вьетнама и его энергетической безопасности с 2015 по 2030 гг. Варианты развития энергетики формировались на основе шести представительных наборов изменений параметров ТЭК для 2020, 2025 и 2030 гг. Изменению подвергался верхний предел возможностей следующих отраслей ТЭК: добыча угля, добыча природного газа, выработка электроэнергии на ГЭС. Перечисленные параметры менялись в границах от -10% до 10% от исходного значения.

Для оценки устойчивого развития национальной энергетики были выбраны 16 индикаторов ЭБ и 24 критерия устойчивого развития. На их базе с помощью метода анализа иерархий [21] были выбраны четыре варианта развития ТЭК. В результате экспертного анализа наилучшим был признан вариант, где в 2020 г. возможности по добыче угля

снижены на 10 %, возможности по добыче природного газа увеличены на 10 %, а в 2025 и 2030 гг. развитие ТЭК идет по базовому варианту.

Используя те же предпосылки и данные, для формирования и анализа вариантов развития ТЭК Вьетнама были применены алгоритмы комбинаторного моделирования.

На первом этапе на основе исходных данных был создан базовый граф развития ТЭК. На втором этапе для сочетаний отраслей ТЭК и регионов Вьетнама, отмеченных знаком + в таблице, были построены графы развития, подобные графу развития объекта, приведенному на рисунке 8. В узлах графа показаны изменения верхнего предела возможностей отрасли ТЭК по отношению к значению, заданному в базовом графе.

Сочетания отраслей ТЭК и регионов Вьетнама

Combining energy sector industries and Vietnam regions

Регион	Добыча угля	Добыча природного газа	Выработка электроэнергии на ГЭС
Дельта Красной реки	+		
Северо-Восточный	+		+
Северо-Западный			+
Север Центрального побережья		+	+
Юг Центрального побережья			+
Центральное нагорье			+
Юго-Восточный		+	+
Дельта Меконга		+	

На третьем этапе был создан граф развития ТЭК, содержащий 531 442 узла. Их расчет на четвертом этапе проводился в распределенной вычислительной среде, состоящей из вычислительных узлов высокопроизводительного вычислительного кластера «Академик В.М. Матросов» [22].

После расчета на модели ТЭК Вьетнама для проверки уровня ЭБ использовались два региональных индикатора ЭБ: доля собственных источников в балансе котельно-печного топлива и доминирующего ресурса в структуре этого топлива [23].

На пятом этапе на основе критерия минимума затрат на развитие и функционирование ТЭК был выбран оптимальный путь развития ТЭК, предусматривающий для всех перспективных моментов времени увеличение добычи природного газа и снижение добычи угля.

Заключение

Важными преимуществами метода комбинаторного моделирования являются представление вариантов развития объектов и затем системы в

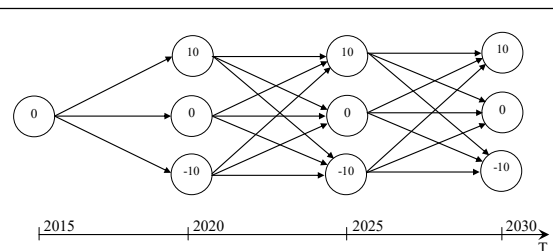


Рис. 8. Граф развития ТЭК Вьетнама

Fig. 8. The graph of Vietnam energy sector development

виде направленного графа, наглядность и компактность в представлении вариантов. Описание в виде графа вариантов наглядно показывает их различия и общие состояния или переходы.

К достоинствам следует отнести достигаемую при этом полноту описания вариантов развития системы. При традиционных способах сравнения вариантов развития, основанных на многокритериальных методах, таких как метод анализа иерархии, исследователи обычно ограничиваются сопоставлением небольшого числа вариантов, выбор которых обосновывается интуицией и накопленным опытом работы. Вместе с тем такой ограниченный выбор, даже если он сделан удачно, всегда отражает некоторую субъективность. Это снижает уровень доказательности получаемых результатов.

Полученные множества допустимых траекторий развития системы могут использоваться во многих постановках задач прогнозирования, в том числе с учетом фактора неопределенности. Среди допустимых траекторий развития системы можно согласно заданным критериям выбирать не только оптимальные пути, но и все множество близлежащих траекторий.

Исследование выполнено при финансовой поддержке РФФИ, проект № 15-07-07412а.

Литература

1. Санеев Б.Г., Соколов А.Д., Агафонов Г.В. Методы и модели разработки региональных энергетических программ. Новосибирск: Наука, 2003. 140 с.
2. Восточный вектор энергетической стратегии России: современное состояние, взгляд в будущее; [под ред. Н.И. Воропая, Б.Г. Санеева]. Новосибирск: Гео, 2011. 368 с.
3. Makarov A.A., Mitrova T.A., Malakhov V.A. World energy forecast and consequences for Russia. Studies on Russian Economic Development, 2013, vol. 24, no. 6, pp. 511–519.
4. Зоркальцев В.И. Методы прогнозирования и анализа эффективности функционирования системы топливоснабжения. М.: Наука, 1988. 144 с.
5. Ibanez E., & McCalley J.D. Multiobjective evolutionary algorithm for long-term planning of the national energy and transportation systems. Energy Systems, 2011, vol. 2, no. 2, pp. 151–169.
6. Веселов Ф.В., Макаров А.А., Малахов В.А. Методы и инструментальный прогнозирования развития электроэнергетики // Изв. РАН. Энергетика. 2010. № 4. С. 82–94.
7. Макаров А.А., Веселов Ф.В., Елисеева О.А. и др. SCANNER. Суперкомплекс активной навигации в энергетических исследованиях. М.: Изд-во ИНЭИ РАН, 2011. 144 с.
8. Fishbone L.G., Abilock H. MARKAL, a linear-program-

ming model for energy systems analysis: technical description of the BNL version. Intern. Jour. of Energy Research, 1981, vol. 5, pp. 353–375.

9. Gerking H. Modeling of multi-stage decision making process in multi-period energy models. Europ. Jour. of Operational Research, 1987, vol. 32, no. 2, pp. 191–204.

10. Van der Voort E., et. al. Energy Supply Modelling Package, EFOM-12C Mark I, Mathematical Description. Louvain-La-Neuve, 1984, 429 p.

11. Loulou R., Labriet M. ETSAP-TIAM: the TIMES integrated assessment model. Part I: Model structure. Computational Management Science, 2008, vol. 5, no. 1, pp. 7–40.

12. Gatautis R., Ravn H.F. Modelling of energy supply systems by the Balmore model. Energetika, 2005, no. 2, pp. 8–20.

13. Зоркальцев В.И., Хамисов О.В. Равновесные модели в экономике и энергетике. Новосибирск: Наука, 2006. 221 с.

14. Береснева Н.М., Еделева А.В. Система поддержки исследований энергетической безопасности России // Программные продукты и системы. 2008. № 2. С. 76–78.

15. Ierusalimsky R., de Figueiredo L.H., Filho W.C. Lua – an extensible extension language. Software-Practice & Experience, 1996, vol. 26, no. 6, pp. 635–652.

16. Edelev A.V., Tchemezov A.V., Nguyen Hoai Nam, Doan Van Binh. Sustainable energy development and green growth strategy for Vietnam: a suitable pathway of power sector ensuring the national energy security. Proc. 3th Intern. Sc. Conf. Sustainable Energy Development, Hanoi, 2013, pp. 21–28.

17. Edelev A.V., Sidorov I.A. Combinatorial modeling approach to find rational ways of energy development with regard to energy security requirements. Proc. 6th Conf. NAA'16. Lozenetz, Bulgaria. 2016, Univ. of Rousse Publ., 16 p.

18. Сидоров И.А., Опарин Г.А., Феоктистов А.Г. Разработка и применение распределенных пакетов прикладных программ // Программные продукты и системы. 2010. № 2. С. 108–111.

19. Edelev A.V., Nguyen Quang Ninh, Nguyen Van The, Tran Viet Hung, Le Tat Tu, Doan Binh Duong, Nguyen Hoai Nam. Developing “Corrective” software: 3-region model. Proc. Intern. Conf. Green Energy and Development, Hanoi, Vietnam, 2012, pp. 41–52.

20. Еделева А.В., Пяткова Н.И., Чemezov А.В., Нгуен Хоай Нам. Программный комплекс «Корректива» для исследований долгосрочного развития топливно-энергетического комплекса Вьетнама // Программные продукты и системы. 2014. № 4. С. 211–216.

21. Sheth N., Hughes L. Quantifying energy security: An Analytic Hierarchy Process approach, Working Paper ERG/200906. Energy Research Group. URL: <http://lh.ece.dal.ca/enen/2009/ERG200906.pdf> (дата обращения: 10.02.2017).

22. Области применения вычислительных ресурсов ИСКЦ. URL: <http://hpc.icc.ru/about/tasks.php> (дата обращения: 10.02.2017).

23. Пяткова Н.И., Рабчук В.И., Сендеров С.М. и др. Энергетическая безопасность России: проблемы и пути решения. Новосибирск: Изд-во СО РАН, 2011. 198 с.

Software & Systems

DOI: 10.15827/0236-235X.118.172-179

Received 13.02.17

2017, vol. 30, no. 2, pp. 172–179

FORMATION OF VIETNAM ENERGY DEVELOPMENT OPTIONS BY COMBINATORIAL MODELING METHODS

A.V. Edelev¹, Ph.D. (Engineering), Senior Researcher, flower@isem.irk.ru

V.I. Zorkaltsev¹, Dr.Sc. (Engineering), Professor, Head of Laboratory, zork@isem.irk.ru

Đoàn Văn Bình², Ph.D. (Engineering), Director, doanbinh@ies.vast.vn

Nguyễn Hoài Nam², Head of Laboratory, nhnam@ies.vast.vn

¹ Melentiev Energy Systems Institute SB RAS, Lermontov St. 130, Irkutsk, 664033, Russian Federation

² Institute of Energy Science of Vietnamese Academy of Science and Technology, 18, Hoàng Quốc Việt, Cầu Giấy, Hà Nội

Abstract. The article describes the combinatorial modelling approach to the research on energy sector development. The idea of approach is to model a system development in the form of a directed graph with nodes corresponding to the possible states of a system at certain moments of time and arcs characterizing the possibility of transitions from one state to another.

The combinatorial modelling is a visual representation of dynamic discrete alternatives. It permits to simulate the long-term process of system development at various possible external and internal conditions, to determine an optimal development strategy of the system under study. The formation and analysis procedures of energy development options are implemented in the Corrective software package. The distributed computing environment are necessary to compute an energy sector development graph.

In 2015, the Institute of Energy Science of the Vietnamese Academy of Science and Technology performed a study of Vietnam sustainable energy development from 2015 to 2030. Data of this study show application of the combinatorial modelling methods to formation and analysis of Vietnam energy development options taking into account energy security requirements. The created Vietnam energy sector development graph consists of 531 442 nodes. It is computed in the cluster located at the Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Science (Irkutsk).

The found optimal way of Vietnam sustainable energy development provides minimum costs of energy sector development and operation.

Keywords: combinatorial modelling, energy development, energy policy, decision support, distributed computing environment.

Acknowledgements. The reported study was funded by RFBR, the research project no. 15-07-07412a.

References

1. Saneev B.G., Sokolov A.D., Agafonov G.V. *Metody i modeli razrabotki regionalnykh energeticheskikh programm* [Methods and Models of Regional Energy Program Development]. Novosibirsk, Nauka Publ., 2003, 140 p.
2. *Vostochny vektor energeticheskoy strategii Rossii: sovremennoe sostoyaniye, vzglyad v budushchee* [The Eastern vector of Russia's energy strategy: the current state, future outlook]. Novosibirsk, "Geo" Publ., 2011, 368 p.
3. Makarov A.A., Mitrova T.A., Malakhov V.A. World energy forecast and consequences for Russia. *Studies on Russian Economic Development*. 2013, vol. 24, no. 6, pp. 511–519.
4. Zorkaltsev V.I. *Metody prognozirovaniya i analiza effektivnosti funktsionirovaniya sistemy toplivsnabzheniya* [The Methods of Forecasting and Analysis of the Efficiency of the Fuel Supply System Operation]. Moscow, Nauka Publ., 1988, 144 p.
5. Ibanez E., McCalley J.D. Multiobjective evolutionary algorithm for long-term planning of the national energy and transportation systems. *Energy Systems*. 2011, vol. 2, no. 2, pp. 151–169.
6. Veselov F.V., Makarov A.A., Malakhov V.A. Methods and tools for predicting the power system development. *Izv. RAN. Energetika* [Proc. of the Russian Academy of Sciences. Power Engineering]. 2010, no. 4, pp. 82–94 (in Russ.).
7. Makarov A.A., Veselov F.V., Eleseeva O.A. *SCANER. Superkompleks aktivnoy navigatsii v energeticheskikh issledovaniyakh* [SCANER. Super Complex of Active Navigation in Energy Researches]. Moscow, ERI RAS Publ., 2011, 144 p.
8. Fishbone L.G., Abilock H. MARKAL, a linear-programming model for energy systems analysis: technical description of the BNL version. *Int. Jour. of Energy Research*. 1981, vol. 5, pp. 353–375.
9. Gerking H. Modeling of multi-stage decision making process in multi-period energy models. *European Jour. of Operational Research*. 1987, vol. 32, no. 2, pp. 191–204.
10. Van der Voort E. *Energy Supply Modelling Package, EFOM-12C Mark I, Mathematical Description*. Louvain-La-Neuve, 1984, 429 p.
11. Loulou R., Labriet M. ETSAP-TIAM: the TIMES integrated assessment model. Part I: Model structure. *Computational Management Science*. 2008, vol. 5, no. 1, pp. 7–40.
12. Gatautis R., Ravn H.F. Modelling of energy supply systems by the Balmorel model. *Energetika*. 2005, no. 2, pp. 8–20.
13. Zorkaltsev V.I., Khamisov O.V. *Ravnovesnye modeli v ekonomike i energetike* [The Equilibrium Model of the Economy and Energy]. Novosibirsk, Nauka Publ., 2006, 221 p.
14. Beresneva N.M., Edelev A.V. A support system of the energy security research in Russia. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 2, pp. 76–78 (in Russ.).
15. Ierusalimsky R., de Figueiredo L.H., Filho W.C. Lua – an extensible extension language. *Software-Practice & Experience*. 1996, vol. 26, no. 6, pp. 635–652.
16. Edelev A.V., Tchemezov A.V., Nguyen H.N., Doan V.B. Sustainable energy development and green growth strategy for Vietnam: a suitable pathway of power sector ensuring the national energy security. *Proc. 3rd Int. Scientific Conf. "Sustainable Energy Development"*. Hanoi, 2013, pp. 21–28.
17. Edelev A.V., Sidorov I.A. Combinatorial Modeling Approach to Find Rational Ways of Energy Development with Regard to Energy Security Requirements. *Proc. 6th Conf. on Numerical Analysis and Applications (NAA'16)*. Lozenetz, Bulgaria. 2016, Univ. of Rousse Publ., 16 p.
18. Sidorov I.A., Oparin G.A., Feoktistov A.G. Development and application of distributed software packages. *Programmnye produkty i sistemy* [Software & Systems]. 2010, no. 2, pp. 108–111 (in Russ.).
19. Edelev A.V., Nguyen Q.N., Nguyen V.T., Tran V. H., Le Tat Tu, Doan B.D., Nguyen H.N. Developing "Corrective" software: 3-region model. *Proc. of Int. Conf. Green Energy and Development*. Hanoi, Vietnam, 2012, pp. 41–52.
20. Edelev A.V., Pyatkova N.I., Chemezov A.V., Nguyen H.N. The software package "Corrective" to study the long-term development of the fuel and energy complex of Vietnam. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4, pp. 211–216.
21. Sheth N., Hughes L. *Quantifying energy security: An Analytic Hierarchy Process approach*. Working Paper ERG/200906, Energy Research Group. Available at: <http://lh.ece.dal.ca/enen/2009/ERG200906.pdf> (accessed February 10, 2017).
22. *Oblasti primeneniya vychislitelnykh resursov ISKTs* [Field of Irkutsk Supercomputer Center Computing Resource Application]. Available at: <http://hpc.icc.ru/about/tasks.php> (accessed February 10, 2017).
23. Pyatkova N.I., Rabchuk V.I., Senderov S.M. *Energeticheskaya bezopasnost Rossii: problemy i puti resheniya* [Energy Security of Russia: Problems and Solutions]. Novosibirsk, SO RAN Publ., 2011, 198 p.

Примеры библиографического описания статьи

1. Еделев А.В., Зоркальцев В.И., Доан Ван Бинь, Нгуен Хоай Нам. Формирование вариантов развития энергетики Вьетнама методами комбинаторного моделирования // Программные продукты и системы. 2017. Т. 30. № 2. С. 172–179; DOI: 10.15827/0236-235X.118.172-179.
2. Edelev A.V., Zorkaltsev V.I., Đoàn Văn Bình, Nguyễn Hoài Nam. Formation of Vietnam energy development options by combinatorial modeling methods. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 172–179 (in Russ.); DOI: 10.15827/0236-235X.118.172-179.

УДК 519.7

DOI: 10.15827/0236-235X.118.180-186

Дата подачи статьи: 06.10.16

2017. Т. 30. № 2. С. 180–186

ПРИНЦИП МИНИМИЗАЦИИ ЭМПИРИЧЕСКОГО РИСКА НА ОСНОВЕ АГРЕГИРУЮЩИХ ФУНКЦИЙ СРЕДНИХ ПОТЕРЬ ДЛЯ РЕШЕНИЯ ЗАДАЧ РЕГРЕССИИ

З.М. Шибзухов, д.ф.-м.н., ведущий научный сотрудник, szport@gmail.ru;

Д.П. Димитриченко, к.т.н., старший научный сотрудник, dimdp@rambler.ru;

М.А. Казаков, младший научный сотрудник, f_wolfgang@mail.ru

*(Институт прикладной математики и автоматизации,
ул. Шортанова, 89а, г. Нальчик, 360000, Россия)*

В настоящей работе предлагается расширенный вариант принципа минимизации эмпирического риска для решения задачи регрессии.

Он строится на основе применения усредняющих агрегирующих функций для вычисления эмпирического риска вместо среднего арифметического. Это оправданно, если распределение потерь имеет выбросы или существенно искажено, отчего оценка риска как средних потерь с самого начала является смещенной. Поэтому в таких случаях при оптимизации параметров в задаче регрессии изначально следует использовать робастную оценку среднего риска.

Подобные оценки среднего риска можно построить, используя усредняющие агрегирующие функции, которые являются решением задачи минимизации штрафной функции за отклонение от своего среднего значения. Такой подход для представления агрегирующих функций среднего позволяет, с одной стороны, определить значительно более широкий класс функций среднего, а с другой, определить дифференцируемые функции среднего, которые аппроксимируют недифференцируемые функции среднего, такие как медиана или квантиль. В результате появляется возможность построить градиентные методы решения задачи регрессии, в определенном смысле аппроксимирующие робастные методы, такие как Least Median и Least Quantile.

В настоящей работе предлагается новая градиентная схема для решения задачи минимизации среднего риска. Она является аналогом схемы, применяемой в алгоритме SAG в случае, когда риск вычисляется при помощи среднего арифметического.

Приведен иллюстративный пример построения робастной процедуры оценки параметров в задаче линейной регрессии на базе использования усредняющей функции среднего, аппроксимирующей медиану.

Ключевые слова: агрегирующая функция, агрегирующая операция, эмпирический риск, регрессия, штрафная функция, процедура градиентного спуска.

Метод минимизации эмпирического риска [1] является признанным методом решения задач параметрической регрессии.

Эмпирический риск обычно вычисляется как среднее арифметическое от значений параметрической функции потерь. Эмпирическая оценка средних потерь как среднее арифметическое адекватна со статистической точки зрения, если потери распределены по нормальному закону. Однако даже для нормального закона среднее арифметическое не является робастной оценкой среднего значения, в то время как медиана позволяет оценивать эмпирическое среднее при наличии выбросов. Поэтому для построения параметрических регрессионных зависимостей также используются эмпирические оценки среднего при помощи медианы, несмотря на то, что использование медианы делает процедуру настройки параметров регрессионной зависимости более медленной.

В условиях выбросов также используют оценки квантилей, когда искажения в распределении потерь составляют меньше 50 %. Это позволяет при настройке параметров при помощи медианы не терять полезную часть распределения потерь, которая расположена выше значения медианы, разделяющей упорядоченный по возрастанию набор потерь на две равные части.

Классический метод эмпирического риска

Задача поиска параметрической регрессионной зависимости $y = f(\mathbf{x}, \mathbf{w})$ между входами \mathbf{x} и скалярным выходом y является одной из классических задач машинного обучения. Имеются конечный набор входов $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_k : k = 1, \dots, N\}$ и набор известных значений на выходе: $\tilde{\mathbf{Y}} = \{\tilde{y}_k : k = 1, \dots, N\}$. Требуется найти такой набор параметров \mathbf{w}^* , при котором функция $f^*(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}^*)$ адекватно представляет зависимость между y и \mathbf{x} на множестве $\tilde{\mathbf{X}}$.

В качестве меры адекватности f^* часто используют эмпирический риск. Набор параметров \mathbf{w}^* , задающий адекватную параметрическую зависимость, должен минимизировать величину эмпирического риска.

Эмпирический риск обычно вычисляется как среднее арифметическое от значений параметрической функции потерь $Q(\mathbf{w}) = \frac{1}{N} \sum_{k=1}^N \ell_k(\mathbf{w})$, где $\ell_k(\mathbf{w}) = \ell_k(r_k(\mathbf{w}))$, где $\ell(r)$ – функция потерь; $r_k(\mathbf{w}) = r(f(\tilde{\mathbf{x}}_k, \mathbf{w}), \tilde{y}_k)$ – функция невязки между значением функции f и ожидаемым значением в k -й точке. Например:

- разность: $r(\mathbf{w}) = f(\mathbf{x}, \mathbf{w}) - y$;

- абсолютная разность: $r(\mathbf{w}) = |f(\mathbf{x}, \mathbf{w}) - y|$;
- несимметричная абсолютная разность: $|f(\mathbf{x}, \mathbf{w}) - y|_a$, где $|r|_a = \begin{cases} \alpha r, & \text{если } r \geq 0, \\ (\alpha - 1)r, & \text{если } r < 0; \end{cases}$
- относительная разность: $r(\mathbf{w}) = \frac{1}{|y|} |f(\mathbf{x}, \mathbf{w}) - y|$

при условии, что значения y отделены от нуля, или

$$r(\mathbf{w}) = \frac{1}{1 + |y|} |f(\mathbf{x}, \mathbf{w}) - y|.$$

Функция потерь – это неотрицательная функция, которая имеет единственный минимум, так что $\ell(0) = \min \ell(r) = 0$. Например:

- абсолютная: $\ell(r) = |r|$;
- квадратичная: $\ell(r) = r^2$;
- Хьюбера: $\ell(r) = \begin{cases} c(2|r| - c), & \text{если } |r| \geq c, \\ r^2, & \text{если } |r| < c; \end{cases}$
- Тьюки (для простоты приведена производная функции):

$$\ell'(r) = \begin{cases} r \left(1 - \left(\frac{r}{c} \right)^2 \right), & \text{если } |r| \leq c, \\ 0, & \text{если } |r| > c; \end{cases}$$

- несимметричная абсолютная:

$$\ell(r) = \begin{cases} \alpha r, & \text{если } r \geq c, \\ (\alpha - 1)r, & \text{если } r < c; \end{cases}$$

- несимметричная квадратичная:

$$\ell(r) = \begin{cases} \alpha r^2, & \text{если } r \geq c, \\ (1 - \alpha)r^2, & \text{если } r < c. \end{cases}$$

Здесь $c > 0$, $0 < \alpha < 1$.

Со статистической точки зрения оценка потерь при помощи среднего арифметического является адекватной, если потери распределены по нормальному закону. Однако если в действительности потери распределены по другому закону, оценка средних потерь должна осуществляться другим способом. Но даже в случае нормально распределенных потерь среднее арифметическое не является устойчивым по отношению к выбросам в эмпирическом распределении. В этом случае существенно более адекватной оценкой является, например, медиана или квантили.

Среднее арифметическое, медиана и квантили – примеры усредняющей агрегирующей функции, поэтому в общем случае средние потери можно вычислять при помощи усредняющих агрегирующих функций.

Усредняющие агрегирующие функции

Пусть $\mathbb{I} \subseteq \mathbb{R}$ – сегмент \mathbb{R} ; \mathbb{I}^* – множество всех конечных последовательностей $\{z_1, \dots, z_N\} \in \mathbb{I}$.

Определение. Агрегирующая функция – это отображение $M: \mathbb{I}^* \rightarrow \mathbb{I}$, которое удовлетворяет следующим требованиям:

- $M\{z\} = z$;
- если $z'_1 \leq z''_1, \dots, z'_N \leq z''_N$, то $M\{z'_1, \dots, z'_N\} \leq M\{z''_1, \dots, z''_N\}$.

Последнее требование – требование монотонности агрегирующей функции.

Агрегирующая функция M симметричная, если $M\{z_1, \dots, z_N\} = M\{z_{\pi(1)}, \dots, z_{\pi(N)}\}$ для любой перестановки π ряда чисел $1, \dots, N$.

Усредняющие агрегирующие функции, по определению, удовлетворяют дополнительному требованию $\min\{z_1, \dots, z_N\} \leq M\{z_1, \dots, z_N\} \leq \max\{z_1, \dots, z_N\}$.

Основные понятия и основные свойства агрегирующих функций подробно описаны в [2–4].

Существует универсальный способ определения усредняющих агрегирующих функций [5]. Для их определения используются штрафные функции.

Определение. Функция $P\{z_1, \dots, z_N, u\}$ является штрафной, если удовлетворяет следующим требованиям:

- $P\{z_1, \dots, z_N, u\} \geq 0$ для всех u и z_1, \dots, z_N ;
- $P\{z_1, \dots, z_N, u\} = 0$, только если $z_1 = \dots = z_N = u$;
- для всех z_1, \dots, z_N множество $M_{z_1, \dots, z_N} = \{u: P(z_1, \dots, z_N, u) = P_{\min}(z_1, \dots, z_N)\}$, где $P_{\min}(z_1, \dots, z_N) = \min_u P(z_1, \dots, z_N, u)$, является синглетоном или связным сегментом.

Всякую усредняющую агрегирующую функцию можно определить на основе некоторой штрафной функции P следующим образом: $M_P\{z_1, \dots, z_N\} = \arg \min_u P(z_1, \dots, z_N, u)$, если M_{z_1, \dots, z_N} – синглетон и $M_P\{z_1, \dots, z_N\} = \frac{a+b}{2}$, если M_{z_1, \dots, z_N} – сегмент с концами a и b .

Заметим, что формально в последнем случае можно было бы выбрать любое значение из интервала (a, b) или некоторое значение из (a, b) , зависящее от P .

Далее рассмотрим разнovidность штрафных функций, которые являются суммами функций несходства:

$$P(z_1, \dots, z_N, u) = \sum_{k=1}^N p(z_k, u), \quad (1)$$

где $p(z, u)$ – функция несходства (dissimilarity function). Функция несходства определяется следующим образом.

Определение. Функция $p(z, u)$ является функцией несходства, если удовлетворяет следующим условиям:

- $p(z, u) = 0 \Leftrightarrow z = u$;
- $p(z_1, u) \geq p(z_2, u)$, когда $z_1 \geq z_2 \geq u$ или $z_1 \leq z_2 \leq u$.

Агрегирующую функцию, определенную на базе штрафной функции вида (1), будем обозначать M_p .

Статистическая интерпретация $M_p\{z_1, \dots, z_N\}$ на основе принципа максимума правдоподобия следующая: если случайная величина z распределена по вероятностному закону $e^{-p(z, \tilde{z})}$, где \tilde{z} – среднее значение, то $M_p\{z_1, \dots, z_N\}$ является эмпирической оценкой \tilde{z} .

Уникальность минимума $P_{z_1, \dots, z_N}(u) = P(z_1, \dots, z_N, u)$ и монотонность $M_p\{z_1, \dots, z_N\}$ гарантированы, когда

$$p(z, u) = G(h(z) - h(u)), \quad (2)$$

где $G: \mathbb{R} \rightarrow \mathbb{R}$ – непрерывная строго выпуклая функция; $h(u)$ – строго монотонная функция [4, 5].

Приведем примеры известных усредняющих агрегирующих функций, которые можно определить таким образом.

- Среднее арифметическое получается при

$$p(z, u) = (z_k - u)^2: M\{z_1, \dots, z_N\} = \arg \max_u \sum_{k=1}^N (z_k - u)^2.$$

- Медиана

$$\text{med}\{z_1, \dots, z_N\} = \begin{cases} z_{(k)}, & \text{если } N = 2k + 1, \\ \frac{z_{(k)} + z_{(k+1)}}{2}, & \text{если } N = 2k, \end{cases}$$

получается при $p(z_k - u) = |z_k - u|$:

$$\text{med}\{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N |z_k - u|,$$

где $z_{(1)}, \dots, z_{(N)}$ – множество z_1, \dots, z_N , упорядоченное в порядке неубывания.

- Квантиль α $Q_\alpha\{z_1, \dots, z_N\}$ получается при $p(z, u) = |z_k - u|_\alpha$:

$$Q_\alpha\{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N |z_k - u|_\alpha,$$

где $|u|_\alpha = \begin{cases} \alpha|u|, & \text{если } u \geq 0, \\ (1-\alpha)|u|, & \text{если } u < 0. \end{cases}$

- Экспектиль α

$$E_\alpha\{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N |z_k - u|_\alpha^2,$$

где $|u|_\alpha^2 = \begin{cases} \alpha u^2, & \text{если } u \geq 0, \\ (1-\alpha)u^2, & \text{если } u < 0. \end{cases}$

- Среднее по Колмогорову

$$M_g\{z_1, \dots, z_N\} = g^{-1}\left(\frac{1}{N} \sum_{k=1}^N g(z_k)\right) \quad \text{получается}$$

при $p(u, z_k) = (g(u) - g(z_k))^2$:

$$M_g\{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N (g(z_k) - g(u))^2.$$

- Масштабированная медиана

$$\text{med}_g\{z_1, \dots, z_N\} = g^{-1}\left(\text{med}\{g(z_k) : k = 1, \dots, N\}\right)$$

получается при $p(u, z_k) = |g(u) - g(z_k)|$:

$$\text{med}_g\{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N |g(z_k) - g(u)|.$$

Поиск значения $M_p\{z_1, \dots, z_N\}$ можно осуществлять методом полного градиента или методом Ньютона. В первом случае на каждом шаге текущая оценка искомого значения обновляется по следующему правилу: $u_{t+1} = u_t - h_t P'_u(u_t, z_1, \dots, z_N)$,

$$\text{где } P'_u(u_t, z_1, \dots, z_N) = \sum_{k=1}^N P'_u(z_k, u_t).$$

Во втором случае обновление осуществляется по правилу

$$u_{t+1} = u_t - h_t \frac{P'_u(u_t, z_1, \dots, z_N)}{P''_{uu}(u_t, z_1, \dots, z_N)},$$

$$\text{где } \frac{P'_u(u_t, z_1, \dots, z_N)}{P''_{uu}(u_t, z_1, \dots, z_N)} = \frac{\sum_{k=1}^N P'_u(z_k, u_t)}{\sum_{k=1}^N P''_{uu}(z_k, u_t)}.$$

Параметр темпа обучения h_t в этих методах может быть постпостоянным или выбираться при помощи одного из методов поиска типа line search.

При больших N удобнее применять стохастические варианты этих алгоритмов. Например, такие алгоритмы, в основе которых лежит такая же схема, как и в основе SAG [6, 7].

В первом случае обновление будет осуществляться по правилу $u_{t+1} = u_t - h_t \bar{g}_t$, где $\bar{g}_t = \frac{1}{N} \sum_{k=1}^N g_{t,k}$, $k(t)$ – номер случайно выбранного значения из z_1, \dots, z_N на шаге t . При этом

$$g_{t+1,k} = \begin{cases} P'_u(z_k, u_t), & \text{если } k = k(t), \\ g_{t,k} & \text{иначе.} \end{cases}$$

Среднее значение производной \bar{g}_t можно обновлять на каждом шаге по простому правилу

$$\bar{g}_{t+1} = \bar{g}_t + \frac{1}{N} (P'_u(z_k, u_t) - g_{t,k}).$$

Во втором случае обновление осуществляется по следующему правилу: $u_{t+1} = u_t - h_t \frac{G_t}{H_t}$, где

$$G_t = \sum_{k=1}^N G_{t,k} \quad \text{и} \quad H_t = \sum_{k=1}^N H_{t,k}.$$

$$\text{При этом } G_{t+1,k} = \begin{cases} P'_u(z_k, u_t), & \text{если } k = k(t), \\ G_{t,k} & \text{иначе,} \end{cases}$$

$$\text{а } H_{t+1,k} = \begin{cases} P''_{uu}(z_k, u_t), & \text{если } k = k(t), \\ H_{t,k} & \text{иначе.} \end{cases}$$

Значение отношения G_t/H_t обновляется на каждом шаге по простому правилу:

$$\frac{G_{t+1}}{H_{t+1}} = \frac{G_t + P'_u(z_k, u_t) - G_{t,k}}{H_t + P''_{uu}(z_k, u_t) - H_{t,k}}.$$

Псевдокод алгоритма – Алг. 1.

Algorithm 1. Алгоритм типа SAG для вычисления значения $M_g\{z_1, \dots, z_N\}$.

Инициализировать u_0

$$\mathbf{g}_k \leftarrow P'(z_k, u_0), k = 1, \dots, N$$

$$G_k \leftarrow P'_u(z_k, u_0), k = 1, \dots, N$$

$$G \leftarrow G_1 + \dots + G_N$$

If используется схема Ньютона **then**

$$H_k \leftarrow p''_{uu}(z_k, u_0), k = 1, \dots, N,$$

$$H \leftarrow H_1 + \dots + H_N$$

end if

$$t \leftarrow 0$$

repeat

$$k \leftarrow k(t)$$

$$G \leftarrow G + p'_u(z_k, u_t) - G_k$$

$$G_k \leftarrow p'_u(z_k, u_t)$$

If используется схема Ньютона **then**

$$H \leftarrow H + p''_{uu}(z_k, u_t) - H_k$$

$$H_k \leftarrow p''_{uu}(z_k, u_t)$$

$$\bar{g} = \frac{G}{H}$$

else

$$\bar{g} = \frac{G}{N}$$

end if

$$u_{t+1} \leftarrow u_t - h_t \bar{g}$$

$$t \leftarrow t + 1$$

until значение u_t не стабилизируется.

От эмпирического риска к агрегированному

Усредняющие агрегирующие функции уже использовались в [8, 9] для построения функционалов потерь в контексте задачи построения операций над алгоритмами классификации и регрессии, которые сохраняют свойство корректности алгоритмов. Применим их теперь для оценки средних потерь: $\mathcal{Q}_p(\mathbf{w}) = M_p\{\ell_k(\mathbf{w}) : k = 1, \dots, N\}$, где усредняющая агрегирующая функция M_p определяется на основе штрафной функции вида (1):

$$M_p\{\ell_1(\mathbf{w}), \dots, \ell_N(\mathbf{w})\} = \arg \min_u \sum_{k=1}^N p(\ell_k(\mathbf{w}), u).$$

Оптимальный набор параметров \mathbf{w}^* доставляет минимум $\mathcal{Q}_p(\mathbf{w})$: $\mathcal{Q}_p(\mathbf{w}^*) = \min_u \mathcal{Q}_p(\mathbf{w})$.

Если $p(z, u)$ имеет частные производные до второго порядка включительно, то

$$\frac{\partial M_p\{z_1, \dots, z_n\}}{\partial z_k} = - \frac{p''_{uz}(z_k, \bar{z})}{\sum_{l=1}^N p''_{uu}(z_l, \bar{z})},$$

где $\bar{z} = M_p\{z_1, \dots, z_n\}$.

Тогда $\text{grad } M_p\{\ell_1(\mathbf{w}), \dots, \ell_N(\mathbf{w})\} =$

$$= \frac{\sum_{k=1}^N p''_{uz}(\ell_k(\mathbf{w}), \bar{z}) \text{grad } \ell_k(\mathbf{w})}{\sum_{k=1}^N p''_{uu}(\ell_k(\mathbf{w}), \bar{z})},$$

где $\bar{z} = M_p\{\ell_1(\mathbf{w}), \dots, \ell_N(\mathbf{w})\}$.

Поиск оптимального набора \mathbf{w} можно осуществлять при помощи следующего варианта про-

цедуры полного градиента. Правило обновления вектора параметров имеет вид

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - h_t \text{grad } M_p\{\ell_1(\mathbf{w}_t), \dots, \ell_N(\mathbf{w}_t)\}.$$

Обновления вектора параметров осуществляются до тех пор, пока значения \mathbf{w}_t и $M_p\{\ell_1(\mathbf{w}_{t+1}), \dots, \ell_N(\mathbf{w}_{t+1})\}$ не стабилизируются.

Если $p(z, u) = G(z - u)$ – частный случай (2), то $\text{grad } M_p\{\ell_1(\mathbf{w}), \dots, \ell_N(\mathbf{w})\} = \sum_{k=1}^N \alpha_k(\mathbf{w}) \text{grad } \ell_k(\mathbf{w})$,

$$\text{где } \alpha_k(\mathbf{w}) = \frac{G''(\ell_k(\mathbf{w}) - \bar{z})}{G''(\ell_1(\mathbf{w}) - \bar{z}) + \dots + G''(\ell_N(\mathbf{w}) - \bar{z})},$$

причем $\alpha_1(\mathbf{w}) + \dots + \alpha_N(\mathbf{w}) = 1$.

Нетрудно заметить, что в этом случае процедура градиентного спуска похожа на процедуру поиска минимума взвешенного среднего от потерь с числовыми весами. Однако в данном случае веса являются функциями от $\ell_1(\mathbf{w}) - \bar{z}, \dots, \ell_N(\mathbf{w}) - \bar{z}$ – отклонений между агрегированным средним от потерь и текущими потерями. Если $G(z - u) = (z - u)^2/2$, то $\alpha_k(\mathbf{w}) = 1/N$, что соответствует среднему арифметическому от потерь или значению эмпирического риска.

Псевдокод алгоритма настройки параметров \mathbf{w} на основе метода полного градиента – алгоритм PBFSG. Приведенный алгоритм не является оптимальным с вычислительной точки зрения, так как на каждом шаге итерации необходимо решать задачу поиска минимума функции для вычисления значения агрегированного среднего значения. Поэтому рассмотрим другой итерационный алгоритм, который ищет значения \mathbf{w}^* и $M_p\{\ell_1(\mathbf{w}^*), \dots, \ell_N(\mathbf{w}^*)\}$ одновременно.

Algorithm PBFSG. Алгоритм полного градиентного спуска на базе агрегирующей функции

$$t \leftarrow 0$$

Инициализировать \mathbf{w}_0

$$u_0 \leftarrow M_p\{\ell_1(\mathbf{w}_0), \dots, \ell_N(\mathbf{w}_0)\}$$

repeat

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - h_t \text{grad } M_p\{\ell_1(\mathbf{w}_t), \dots, \ell_N(\mathbf{w}_t)\}$$

$$u_{t+1} \leftarrow M_p\{\ell_1(\mathbf{w}_{t+1}), \dots, \ell_N(\mathbf{w}_{t+1})\}$$

$$t \leftarrow t + 1$$

until $\{u\}$ и $\{\mathbf{w}_t\}$ не стабилизируются.

Алгоритм стохастического усредненного градиента на базе агрегирующих функций

Поскольку данный усредненный градиент является взвешенной суммой градиентов от соответствующих потерь, можно применить метод, лежащий в основе алгоритма SAG (Stochastic Average Gradient) [6, 7]. Построим на основе этого метода алгоритм PBSAG (Penalty Based Stochastic Average Gradient) стохастически усредненного градиента

на базе усредняющей верной агрегирующей функции. Схема адаптации параметров \mathbf{w} и u имеет вид

$$\mathbf{w}_{t+1} = \mathbf{w}_t - h_t \bar{\mathbf{g}}_t, \quad u_{t+1} = u_t - \tau_t \bar{q}_t,$$

$$\text{где } \bar{\mathbf{g}}_t = \frac{\sum_{k=1}^N \mathbf{g}_{k,t}}{\sum_{k=1}^N \mathbf{g}_{k,t}}.$$

Значение \bar{q}_t для поиска минимального значения усредняющей агрегирующей функции M_p может обновляться в соответствии с одним из следующих правил:

$$\bar{q}_t = \frac{1}{N} \sum_{k=1}^N q_{k,t} \quad \text{или} \quad \bar{q}_t = \frac{\sum_{k=1}^N q_{k,t}}{\sum_{k=1}^N \mathbf{g}_{k,t}} \quad \text{в зависимости от используемого метода: градиентного спуска или Ньютона. Векторы из набора } \{\mathbf{g}_{k,t} : k = \overline{1, N}\} \text{ обновляются по следующему правилу:}$$

$$\mathbf{g}_{k,t+1} = \begin{cases} -p_{uz}''(\ell_k(\mathbf{w}_t), u_t) \text{grad } \ell_k(\mathbf{w}_t), & \text{если } k = k(t), \\ \mathbf{g}_{k,t} & \text{иначе.} \end{cases}$$

Значения из наборов $\{\mathbf{g}_{k,t} : k = \overline{1, N}\}$ и $\{q_{k,t} : k = \overline{1, N}\}$ обновляются по следующим правилам:

$$\mathbf{g}_{k,t+1} = \begin{cases} p_{uu}''(\ell_k(\mathbf{w}_t), u_t), & \text{если } k = k(t), \\ \mathbf{g}_{k,t} & \text{иначе,} \end{cases}$$

$$q_{k,t+1} = \begin{cases} p_u'(\ell_k(\mathbf{w}_t), u_t), & \text{если } k = k(t), \\ q_{k,t} & \text{иначе.} \end{cases}$$

Algorithm PBSAG. Алгоритм стохастически усредненного градиента на базе усредняющей функции

$t \leftarrow 0$

Инициализировать \mathbf{w}_0

for $k \in \{1, \dots, N\}$ **do**

$$\mathbf{G}_k \leftarrow p_{uz}''(\ell_k(\mathbf{w}_0), u_0) \text{grad } \ell_k(\mathbf{w}_0)$$

$$\mathbf{H}_k \leftarrow p_{uu}''(\ell_k(\mathbf{w}_0), u_0)$$

$$Q_k \leftarrow p_u'(\ell_k(\mathbf{w}_0), u_0)$$

end for

$$\mathbf{G} \leftarrow \mathbf{G}_1 + \dots + \mathbf{G}_N$$

$$\mathbf{H} \leftarrow \mathbf{H}_1 + \dots + \mathbf{H}_N$$

$$Q \leftarrow Q_1 + \dots + Q_N$$

repeat

$$k = k(t)$$

$$\mathbf{G} \leftarrow \mathbf{G} - \mathbf{G}_k + p_{uz}''(\ell_k(\mathbf{w}_0), u_t) \text{grad } \ell_k(\mathbf{w}_t)$$

$$\mathbf{G}_k \leftarrow p_{uz}''(\ell_k(\mathbf{w}_0), u_t) \text{grad } \ell_k(\mathbf{w}_t)$$

$$\mathbf{H} \leftarrow \mathbf{H} - \mathbf{H}_k + p_{uu}''(\ell_k(\mathbf{w}_0), u_t)$$

$$\mathbf{H}_k \leftarrow p_{uu}''(\ell_k(\mathbf{w}_0), u_t)$$

$$Q \leftarrow Q - Q_k + p_u'(\ell_k(\mathbf{w}_0), u_t)$$

$$Q_k \leftarrow p_u'(\ell_k(\mathbf{w}_0), u_t)$$

$$\bar{\mathbf{g}} = \frac{\mathbf{G}_1}{\mathbf{G}_2}$$

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - h_t \bar{\mathbf{g}}$$

if используется схема Ньютона **then**

$$\bar{q} \leftarrow Q / \mathbf{G}_2$$

else

$$\bar{q} \leftarrow Q / N$$

end if

$$u_{t+1} \leftarrow u_t - \tau_t \bar{q}$$

$$t \leftarrow t + 1$$

until $\{u_t\}$ и $\{\mathbf{w}_t\}$ не стабилизируются

Алгоритму PBSAG на каждом шаге необходимо хранить по одному градиентному вектору и по два значения на каждый пример из обучающего набора данных, то есть $N(m+2)$ вещественных чисел, где m – ранг вектора параметров \mathbf{w} . Поэтому его следует применять, если есть память для хранения такого объема данных.

Нетрудно заметить, что при $p(z, u) = (z-u)^2/2$ схема алгоритма PBSAG редуцируется к схеме алгоритма SAG: $\mathbf{w}_{t+1} = \mathbf{w}_t - h_t \mathbf{g}_t$, где $\mathbf{g}_t = \frac{1}{N} \sum_{k=1}^N \mathbf{g}_{k,t}$,

$$\mathbf{g}_{k,t+1} = \begin{cases} \text{grad } \ell_k(\mathbf{w}_t), & \text{если } k = k(t), \\ \mathbf{g}_{k,t} & \text{иначе.} \end{cases}$$

Таким образом, схема алгоритма PBSAG является естественным обобщением схемы алгоритма SAG [6, 7], когда для вычисления средних потерь используется усредняющая агрегирующая функция, основанная на штрафной, вместо среднего арифметического.

Примеры применения PBSAG

Рассмотрим применение PBSAG с использованием аппроксимированного варианта LMS [10, 11] для поиска линейной регрессии в условиях выбросов. В стандартном алгоритме LMS ищется минимум медианы квадрата ошибки:

$$\mathcal{E}(\mathbf{w}) = \text{med} \left\{ (f(\mathbf{x}_k, \mathbf{w}) - y_k)^2 : k = 1, \dots, N \right\}.$$

PBSAG нельзя применить, когда M является медианой. Однако ее можно заменить на суррогат медианы, который асимптотически эквивалентен ей.

Определение. $p_\alpha(z-u)$ определяет суррогат медианы, асимптотически эквивалентный медиане, если для некоторого α^*

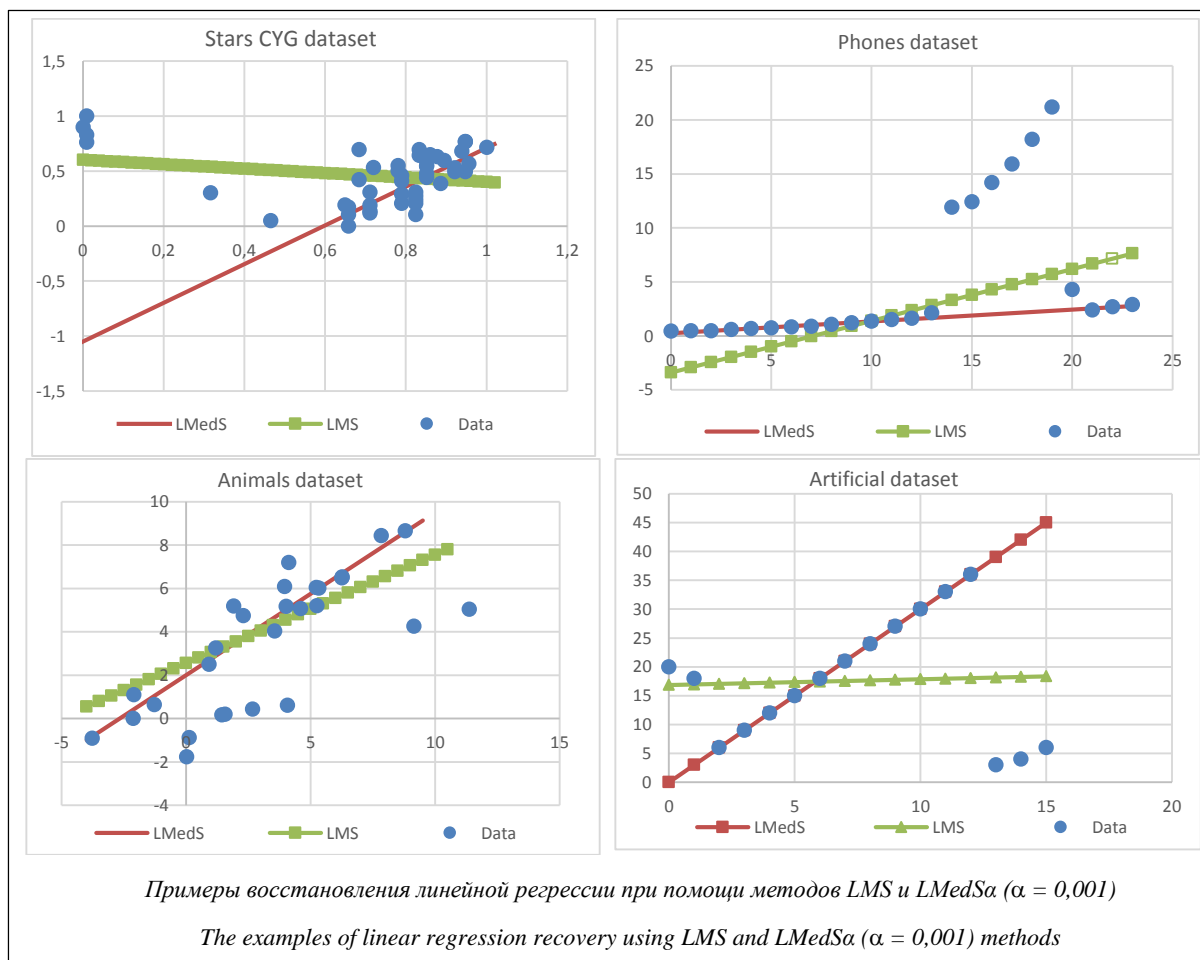
$$\lim_{\alpha \rightarrow \alpha^*} p_\alpha(z-u) = |z-u|,$$

$$\lim_{\alpha \rightarrow \alpha^*} p_\alpha'(z-u) = \text{sign}(z-u).$$

Рассмотрим пример:

$$p_\alpha(z-u) = \sqrt{\alpha^2 + |z-u|^2} - \alpha,$$

где $\alpha^* = 0$,



$$p'_\alpha(z-u) = \frac{z-u}{(\alpha^2 + |z-u|^2)^{1/2}},$$

$$p''_\alpha(u-z) = \frac{\alpha^2}{(\alpha^2 + |z-u|^2)^{3/2}}.$$

Соответствующую ей усредняющую агрегирующую функцию будем называть α -медианой:

$$\text{med}_\alpha \{z_1, \dots, z_N\} = \arg \min_u \sum_{k=1}^N \left(\sqrt{\alpha^2 + |z_k - u|^2} - \alpha \right).$$

Другой пример суррогата можно построить на основе следующей функции несходства:

$$p_\alpha(z-u) = |z-u| - \alpha \ln(\alpha + |z-u|) + \alpha \ln \alpha. \quad (3)$$

На рисунке представлены примеры применения алгоритма PBSAG. Они показывают способность метода и алгоритма PBSAG на базе усредняющего верного агрегирующего функционала, аппроксимирующего медиану (3), восстанавливать линейную регрессионную зависимость в случае, когда исходные данные содержат выбросы.

Заключение

В данной работе эти функции используются для оценки средних потерь, где усредняющая агрегиру-

ющая функция определяется на основе штрафной функции. В результате процедура градиентного спуска становится аналогичной процедуре поиска минимума взвешенного среднего от потерь с числовыми весами. Это позволяет построить алгоритм PBSAG – стохастически усредненного градиента на базе усредняющей верной агрегирующей функции. Предложенный алгоритм, реализующий метод минимизации эмпирического риска, позволяет справляться с задачей восстановления линейной регрессионной зависимости в случае, когда исходные данные содержат выбросы. Данное свойство алгоритма продемонстрировано на соответствующих примерах.

(Работа выполнена при поддержке гранта РФФИ № 15-01-03381 и гранта ОННТ РАН).

Литература

1. Vapnik V. The nature of statistical learning theory (Information Science and Statistics). Springer-Verlag. NY, 2000, 314 p.
2. Mesiar R., Komornikova M., Kolesarova A., Calvo T. Aggregation functions: a revision. In H. Bustince, F. Herrera, J. Montero, eds. (Fuzzy Sets and Their Extensions: Representation, Aggregation and Models). Springer, Berlin, Heidelberg, 2008.
3. Grabich M., Marichal J.-L., Pap E. Aggregation functions (Encyclopedia of Mathematics and its Applications), Cambridge Univ. Press, 2009, no. 127.
4. Beliakov G., Sola H., Calvo T. A practical guide to averaging functions. Springer, 2016, 329 p.

5. Calvo T., Beliakov G. Aggregation functions based on penalties. *Fuzzy Sets and Systems*. 2010, vol. 161, no. 10, pp. 1420–1436.
6. Le Roux N., Schmidt M., Bach F. A stochastic gradient method with an exponential convergence rate for finite training sets. 2012. URL: <http://arxiv.org/abs/1202.6258> (дата обращения: 05.10.2016).
7. Schmidt M., Le Roux N., Bach F. Minimizing finite sums with the stochastic average gradient. 2013. URL: <http://arxiv.org/abs/1309.2388> (дата обращения: 05.10.2016).
8. Shibzukhov Z.M. Correct aggregate operations with algorithms, *Pattern Recognition and Image Analysis*, 2014, vol. 24, no. 3, pp. 377–382.
9. Shibzukhov Z.M. Aggregation correct operations on algorithms. *Dokl. Math.* 2015, vol. 91, no. 3, pp. 391–393.
10. Rousseeuw P.J. Least median of squares regression. *Jour. of the American Statistical Association*, 1984, no. 79, pp. 871–880.
11. Rousseeuw P.J., Leroy A.M. *Robust regression and outlier detection*. NY, John Wiley and Sons, 1987.

Software & Systems
DOI: 10.15827/0236-235X.118.180-186

Received 06.10.16
2017, vol. 30, no. 2, pp. 180–186

THE EMPIRICAL RISK MINIMIZATION PRINCIPLE BASED ON AVERAGE LOSS AGGREGATING FUNCTIONS FOR REGRESSION PROBLEMS

Z.M. Shibzukhov¹, *Dr.Sc. (Physics and Mathematics), Leading Researcher, szport@gmail.ru*
D.P. Dimitrichenko¹, *Ph.D. (Engineering), Senior Researcher, dimdp@rambler.ru*
M.A. Kazakov¹, *Junior Researcher, f_wolfgang@mail.ru*

¹ *Institute of Applied Mathematics and Automation, Shortanova St. 89a, Nalchik, 360000, Russian Federation*

Abstract. The paper proposes an extended principle of empirical risk minimization to solve the regression problem. It is based on using aggregate functions instead of arithmetic mean to calculate risk. This can be justified if the loss distribution of emissions is significant or distorted, causing a shift in the risk assessment of the average loss from the very beginning. Therefore, in such cases, when optimizing characteristics in the regression problem the robust estimate of average value-at-risk should be initially used.

Such intermediate risk assessment can be constructed using avg functions, which are the solution to the problem of penalty function minimization in case of mean deviation. This approach allows, on one hand, to determine a much broader class of secondary functions, and, on the other hand, to determine the average differentiable functions that approximate the average non-differentiable functions, such as a median or quintile. As a result, it is possible to construct gradient methods for solving the regression problem that, in a sense, can approximate robust techniques such as Least Median and Least Quintile.

This paper proposes a new gradient scheme for solving the minimization problem of the intermediate risk. It is an analog of the used in the SAG algorithm circuit when the risk is calculated by arithmetic mean. An illustrative example presents the construction of robust procedures for characteristics assessment in a linear regression based on the use of the avg function, which approximates the median.

Keywords: aggregation function/operation, empirical risk, regression, penalty function, gradient descent procedure.

Acknowledgements. This work is supported by RFBR grant no. 15-01-03381 and RAS Onita grant.

References

1. Vapnik V. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag Publ., 2000.
2. Mesiar R., Komornikova M., Kolesarova A., Calvo T. Aggregation functions: A revision. H. Bustince, F. Herrera, J. Montero (Eds.). *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*. Springer, Berlin, Heidelberg Publ., 2008.
3. Grabich M., Marichal J.-L., Pap E. *Aggregation Functions*. Series: Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, 2009, no. 127.
4. Beliakov G., Sola H., Calvo T. *A Practical Guide to Averaging Functions*. 2016, Springer Publ., 329 p.
5. Calvo T., Beliakov G. Aggregation functions based on penalties. *Fuzzy Sets and Systems*. 2010, vol. 161, no. 10, pp. 1420–1436.
6. Le Roux N., Schmidt M., Bach F. *A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets*. nips.org, 2012. Available at: <http://arxiv.org/abs/1202.6258> (accessed October 5, 2016).
7. Schmidt M., Le Roux N., Bach F. *Minimizing Finite Sums with the Stochastic Average Gradient*. arXiv.org, 2013. Available at: <http://arxiv.org/abs/1309.2388> (accessed October 5, 2016).
8. Shibzukhov Z.M. Correct Aggregate Operations with Algorithms. *Pattern Recognition and Image Analysis*. 2014, vol. 24, no. 3, pp. 377–382.
9. Shibzukhov Z.M. Aggregation correct operations on algorithms. *Dokl. Math.* 2015, vol. 91, no. 3, pp. 391–393.
10. Rousseeuw P.J. Least Median of Squares Regression. *Jour. of the American Statistical Association*. 1984, no. 79, pp. 871–880.
11. Rousseeuw P.J., Leroy A.M. *Robust Regression and Outlier Detection*. NY, John Wiley and Sons Publ., 1987.

УДК 519.7

DOI: 10.15827/0236-235X.118.187-195

Дата подачи статьи: 17.03.17

2017. Т. 30. № 2. С. 187–195

СЖАТОЕ ПРЕДСТАВЛЕНИЕ СТРОГИХ АССОЦИАТИВНЫХ ПРАВИЛ В АНАЛИЗЕ ДАННЫХ

В.В. Быкова, д.ф.-м.н., профессор, bykvalen@mail.ru;

А.В. Катаева, аспирант, kataeva_av@mail.ru

*(Сибирский федеральный университет,
Свободный просп., 79, г. Красноярск, 660041, Россия)*

Современные методы и средства поиска ассоциативных правил в больших массивах данных приводят к значительному множеству правил, многие из которых являются избыточными. Избыточные ассоциативные правила не представляют ценности, но могут вводить в заблуждение. Для решения этой проблемы предложен алгоритм MClose, являющийся модификацией алгоритма Close.

Известно, что с помощью алгоритма Close можно построить минимаксный базис для строгих ассоциативных правил (правил с достоверностью 1). Минимаксный базис интересен для экспертов тем, что каждое входящее в него правило имеет минимальную посылку и максимальное следствие. Однако минимаксный базис может содержать избыточные ассоциативные правила. Алгоритм MClose позволяет в процессе построения минимаксного базиса устранять избыточные правила. Результирующий базис назван сжатым строгим базисом. Удаленные ассоциативные правила всегда можно получить из сжатого строгого базиса с сохранением их поддержки и достоверности без обращений к анализируемому массиву данных. Алгоритм MClose основан на соответствиях Галуа и выводимостях, подобных аксиомам Амстронга, которые используются в теории реляционных БД для функциональных зависимостей.

Как показали вычислительные эксперименты, алгоритм MClose по времени работы сопоставим с алгоритмом Close. Однако он примерно в два раза уменьшает число ассоциативных правил минимаксного базиса. В работе дано описание программы, в которой представлены алгоритмы MClose и Close.

Ключевые слова: анализ данных, соответствия Галуа, замкнутые множества, ассоциативные правила, избыточность, сжатый строгий базис.

Интеллектуальный анализ данных направлен на исследование больших объемов информации с целью выявления зависимостей между данными. Ассоциативные правила (*association rule*) – один из хорошо изученных классов зависимостей, которые отражают, какие признаки или события встречаются совместно и насколько часто это происходит. В настоящее время ассоциативные правила нашли широкое применение в медицине, информационной безопасности, анализе компьютерных сетей и маркетинге при решении задач диагностики и прогнозирования [1–3].

При поиске ассоциативных правил анализируемое множество данных обычно описывается бинарным контекстом – матрицей, строки которой соответствуют объектам рассматриваемой предметной области, а столбцы – признакам этих объектов. Единичное значение элемента матрицы трактуется как наличие у объекта соответствующего признака, а нулевое – как его отсутствие. Бинарное представление данных значительно расширяет математический аппарат для их исследования. Современные методы поиска ассоциативных правил базируются преимущественно на анализе формальных понятий и теории вероятностей [4, 5]. Анализ формальных понятий, как прикладная ветвь алгебраической теории решеток, является удобным математическим аппаратом описания методов поиска ассоциативных правил [6–8].

Главная проблема при поиске ассоциативных правил – огромное число правил, возникающих при исследовании больших бинарных контекстов.

Это существенно усложняет экспертный анализ выявленных ассоциативных правил. Для решения этой проблемы используются различные меры значимости, такие как поддержка (*support*) и достоверность (*confidence*) [9]. С их помощью найденные ассоциативные правила фильтруются, и для анализа предъявляются только те, для которых значения мер значимости превышают заданные пороговые значения. Подобная фильтрация, конечно, уменьшает число правил, но не решает проблему размерности полностью. Не помогает и расширение мер значимости [10]. Часто после фильтрации все равно остается значительное число ассоциативных правил, при этом многие из них избыточные. Ассоциативное правило считается избыточным, если его удаление из множества правил не приводит к потере информации о связях между данными в рассматриваемой предметной области.

Существуют различные формальные определения избыточных ассоциативных правил и методы их устранения [11]. Наиболее развиты методы устранения избыточности для строгих ассоциативных правил (*strong association rules*). Такие правила имеют достоверность 1 и считаются самыми важными. Для этих правил имеются плотная параллель с функциональными зависимостями из теории реляционных БД, а также четкое понятие избыточности [12]. Строгие ассоциативные правила представляют интерес для тех приложений, где требуется высокая степень уверенности в обнаруженных зависимостях между данными, например, в медицине и информационной безопасности [1–3].

Множество ассоциативных правил, не содержащее избыточных (в некотором смысле) правил, принято называть базисом. В настоящее время известен ряд алгоритмов, позволяющих строить различные базисы для строгих ассоциативных правил. Наиболее значимыми являются канонический и минимаксный базисы. Канонический базис (или базис Дюкена–Гига) состоит из минимального числа ассоциативных правил, рекуррентно описываемых в терминах псевдосодержаний [13]. Канонический базис глубоко исследован в анализе формальных понятий, однако все предложенные на сегодняшний день алгоритмы его построения в большей степени представляют теоретический, чем практический интерес [14]. Минимаксный базис состоит из строгих ассоциативных правил, имеющих минимальную посылку и максимальное следствие [15]. Для построения минимаксного базиса существует ряд хорошо апробированных практикой алгоритмов. К ним относится, например, алгоритм Close [11, 16]. Исследования показали, что в построенных канонических и минимаксных базисах остается некоторая избыточность, которая может быть устранена на основе выводимостей, подобных аксиомам Амстронга, известным в теории реляционных БД для функциональных зависимостей.

В данной статье для строгих ассоциативных правил предложен алгоритм MClose – модификация известного алгоритма Close. Алгоритм MClose позволяет в процессе построения минимаксного базиса устранять избыточность с сохранением поддержки и достоверности без дополнительного обращения к исходному бинарному контексту. Результирующий базис назван сжатым строгим базисом. Введено понятие избыточного ассоциативного правила. Приведены выводимости, доказывающие корректность предлагаемого алгоритма, а также результаты вычислительных экспериментов. Поскольку алгоритм Close основан на анализе формальных понятий, необходимые определения и обозначения этой теории возьмем из работ [6–8].

Основные определения и обозначения анализа формальных понятий

Пусть для некоторой предметной области определены два непустых конечных множества G и M , объектов и признаков соответственно. Предполагаем, что все объекты в G и признаки в M различны. Пусть задано отношение $I \subseteq G \times M$ инцидентности между множествами G и M . Существование в I пары (g, m) , $g \in G$ и $m \in M$, означает, что объект g имеет признак m и, наоборот, признак m характерен для объекта g . Тройку $K = (G, M, I)$ принято называть контекстом предметной области.

Выберем в $K = (G, M, I)$ два любых элемента $g \in G$ и $m \in M$. Определим для них два отображения φ и ψ :

$\varphi(g) = \{m \in M \mid (g, m) \in I\}$ – множество признаков, присущих объекту g ;

$\psi(m) = \{g \in G \mid (g, m) \in I\}$ – множество объектов, обладающих признаком m .

Отображения φ и ψ обобщаются на произвольные множества $A \subseteq G$ и $B \subseteq M$ следующим образом:

$$\varphi(A) = \bigcap_{g \in A} \varphi(g) = \{m \in M \mid \forall g \in A (g, m) \in I\},$$

$$\psi(B) = \bigcap_{m \in B} \psi(m) = \{g \in G \mid \forall m \in B (g, m) \in I\}.$$

Следовательно, $\varphi(A)$ – множество признаков, общих для всех объектов из A , а $\psi(B)$ – множество объектов, которые обладают всеми признаками из B . Полагаем, что $\varphi(\emptyset) = M$ и $\psi(\emptyset) = G$: пустому множеству объектов присущи все признаки из M и каждый объект рассматриваемого контекста $K = (G, M, I)$ обладает пустым множеством признаков.

Обычно в анализе формальных понятий для отображений φ и ψ применяется единое обозначение $(\cdot)'$, а приведенные выше формулы для $\varphi(A)$, $\psi(B)$ записываются так:

$$A' = \bigcap_{g \in A} g' = \{m \in M \mid \forall g \in A (g, m) \in I\}, \quad (1)$$

$$B' = \bigcap_{m \in B} m' = \{g \in G \mid \forall m \in B (g, m) \in I\}. \quad (2)$$

Если $g \in G$ и $m \in M$, то обозначения g' и m' служат сокращенной формой записи множеств $\varphi(g) = \{g'\}$ и $\psi(m) = \{m'\}$ соответственно.

Отображения «'» удовлетворяют свойствам, вытекающим из их определения и вполне реалистичного и постулируемого в анализе данных положения: расширение (сокращение) множества признаков уменьшает (увеличивает) число объектов, обладающих этими признаками. Формально эти свойства можно выразить в виде следующих утверждений.

Утверждение 1. Для всякого контекста $K = (G, M, I)$ и любых $B_1, B_2 \subseteq M$ верны свойства:

– *антимонотонность*:

если $B_1 \subseteq B_2$, то $(B_2)' \subseteq (B_1)'$;

– *экстенсивность*:

$B_1 \subseteq (B_1)''$, где $(B_1)'' = ((B_1)')' \subseteq M$.

Множество $(B_1)'' = \varphi(\psi(B_1))$ можно трактовать как набор признаков, которые всегда появляются в объектах контекста $K = (G, M, I)$ вместе с признаками из B_1 , причем это множество является наибольшим по включению в пределах $K = (G, M, I)$. Очевидно, что $(\emptyset)'' = \varphi(\psi(\emptyset)) = G'$, где G' – множество признаков, свойственных всем объектам из G . Для подмножеств множества G справедливо утверждение 2, аналогичное утверждению 1.

Утверждение 2. Для всякого контекста $K = (G, M, I)$ и любых $A_1, A_2 \subseteq G$ верны свойства:

– *антимонотонность*:

если $A_1 \subseteq A_2$, то $(A_2)' \subseteq (A_1)'$;

– *экстенсивность*:

$A_1 \subseteq (A_1)''$, где $(A_1)'' = ((A_1)')' \subseteq G$.

Множество $(A_1)'' = \psi(\varphi(A_1))$ можно интерпретировать как наибольшее по включению множество

объектов, обладающих всеми признаками, которые имеют объекты из A_1 .

Заметим, что, согласно утверждениям 1 и 2, отображения φ и ψ составляют пару соответствий Галуа между множествами 2^G и 2^M , частично упорядоченными по включению [6, 7]. Здесь традиционно 2^G и 2^M – совокупность всех подмножеств рассматриваемых множеств G и M соответственно. Известно, что для соответствий Галуа [7] φ и ψ справедливо равенство

$$\varphi(\psi(\varphi(A))) = \varphi(A), \varphi(\psi(\psi(B))) = \psi(B),$$

или в единых обозначениях

$$((A')') = (A')' = A', ((B')') = (B')' = B'.$$

Двойное применение отображения «'» определяет оператор замыкания на 2^M в алгебраическом смысле [8]. Ему присущи

– *рефлексивность*:

для любого $B \subseteq M$ всегда $B \subseteq B''$;

– *монотонность*:

если $B_1 \subseteq B_2 \subseteq M$, то $(B_1)'' \subseteq (B_2)'' \subseteq M$;

– *идемпотентность*:

для любого $B \subseteq M$ всегда $(B'')'' = B''$.

Справедливость этих свойств вытекает из утверждений 1 и 2. Множество признаков $B \subseteq M$, для которого $B = B''$, называется замкнутым относительно оператора «''» в контексте $K = (G, M, I)$. Принято говорить, что множество B'' является замыканием для $B \subseteq M$ в контексте $K = (G, M, I)$. Заметим, что, исходя из (1) и (2), при $B' \neq \emptyset$ замыкание для $B \subseteq M$ можно вычислить по формуле

$$B'' = \bigcap_{g \in G} \{g' \mid B \subseteq g'\}. \quad (3)$$

Если $B' = \emptyset$, то всегда $B'' = \varphi(\psi(B)) = \varphi(\emptyset) = M$.

Важно отметить, что применение формулы (3) позволяет найти для заданного множества признаков замыкание за один просмотр контекста $K = (G, M, I)$.

Ассоциативные правила и основные меры их значимости

Определим ассоциативное правило, используя приведенные выше понятия и обозначения. Ассоциативным правилом на множестве признаков контекста $K = (G, M, I)$ называется упорядоченная пара $r = (X, Y)$, $X, Y \subseteq M$.

Принято ассоциативное правило $r = (X, Y)$ записывать в виде $X \Rightarrow Y$. В ассоциативном правиле $X \Rightarrow Y$ множества X и Y называют посылкой (или причиной) и заключением (или следствием) соответственно [9]. В анализе ассоциативных правил часто полагают, что посылка и заключение – непустые непересекающиеся множества. С формальных позиций эти ограничения несущественны.

Применительно к заданному контексту $K = (G, M, I)$ всякое ассоциативное правило $X \Rightarrow Y$ количественно характеризуется с помощью двух числовых функций: $\delta(X \Rightarrow Y)$ – поддержка, $\gamma(X \Rightarrow Y)$ – достоверность [10]. Поддержка и достоверность ассо-

циативного правила определяются через понятие поддержки множества признаков.

Поддержка $\delta(X)$ множества признаков $X \subseteq M$ в контексте $K = (G, M, I)$ – отношение числа объектов, которым присущи признаки X , к общему числу объектов, представленных в этом контексте:

$$\delta(X) = |X'| / |G|. \quad (4)$$

Величину $\delta(X)$ можно интерпретировать как частоту встречаемости в контексте $K = (G, M, I)$ объектов, имеющих признаки X . Из формулы (4) следует, что для любого $X \subseteq M$ значение $\delta(X)$ неизменно находится в естественных границах $0 \leq \delta(X) \leq 1$. Чем ближе значение $\delta(X)$ к 1, тем большее число объектов рассматриваемого контекста обладает всеми признаками из X . В силу антимонотонности отображения «'» поддержка множества признаков также удовлетворяет свойству антимонотонности: для всякого контекста $K = (G, M, I)$ и любых $X, Y \subseteq M$ при $X \subseteq Y$ верно неравенство

$$\delta(Y) \leq \delta(X). \quad (5)$$

Согласно (5), поддержка множества признаков не может превышать поддержку любого из его подмножеств. Так, для произвольного $X \subseteq M$ всегда $\delta(M) \leq \delta(X) \leq \delta(\emptyset) = 1$.

Множество признаков $X \subseteq M$ называется частым в контексте $K = (G, M, I)$, если его поддержка больше или равна заданному пороговому значению $\delta_0 \in [0, 1]$. Если $\delta(X) \geq \delta_0$ и $X = X''$, то X называется частым замкнутым множеством признаков в $K = (G, M, I)$. Частые множества и частые замкнутые множества признаков традиционно служат основой для поиска ассоциативных правил в заданном контексте.

Поддержкой $\delta(X \Rightarrow Y)$ ассоциативного правила $X \Rightarrow Y$ относительно контекста $K = (G, M, I)$ называется величина

$$\delta(X \Rightarrow Y) = \delta(X \cup Y) = |(X \cup Y)'| / |G|, \quad (6)$$

указывающая, какая доля объектов этого контекста имеет признаки $X \cup Y$. Достоверность $\gamma(X \Rightarrow Y)$ ассоциативного правила $X \Rightarrow Y$ относительно контекста $K = (G, M, I)$ определяется как отношение числа объектов, обладающих всеми признаками из $X \cup Y$, к числу объектов, которым свойственны только признаки X : $\gamma(X \Rightarrow Y) = |(X \cup Y)'| / |X'|$.

Достоверность ассоциативного правила через функцию поддержки выражается формулой

$$\gamma(X \Rightarrow Y) = \delta(X \Rightarrow Y) / \delta(X) = \delta(X \cup Y) / \delta(X). \quad (7)$$

Заметим, что достоверность определяется формулой (7) только для тех ассоциативных правил $X \Rightarrow Y$, для которых $\delta(X) \neq 0$. Если $\delta(X) = 0$ (в контексте нет ни одного объекта, который обладал бы признаками X), то также $\delta(X \cup Y) = 0$. В этом особом случае полагают $\gamma(X \Rightarrow Y) = 1$. Исходя из (4)–(7), достоверность ассоциативного правила $X \Rightarrow Y$ при произвольных $X, Y \subseteq M$ всегда находится в границах $0 \leq \gamma(X \Rightarrow Y) \leq 1$. Чем ближе значение $\gamma(X \Rightarrow Y)$ к 1, тем с большей уверенностью

можно сказать, что признаки Y появляются в объектах рассматриваемого контекста вместе с признаками X .

Ассоциативное правило $X \Rightarrow Y$ называется минимаксным, если не существует другого ассоциативного правила $X^* \Rightarrow Y^*$, такого, что $X^* \subseteq X$, $Y \subseteq Y^*$ и $\delta(X^* \Rightarrow Y^*) = \delta(X \Rightarrow Y)$, $\gamma(X^* \Rightarrow Y^*) = \gamma(X \Rightarrow Y)$.

В анализе ассоциативных правил часто достоверность ассоциативного правила называют величиной его значимости. Самыми значимыми считаются строгие ассоциативные правила – правила с достоверностью 1. Известен критерий наличия в контексте строгого ассоциативного правила: достоверность ассоциативного правила $X \Rightarrow Y$ относительно контекста $K = (G, M, I)$ равна 1 тогда и только тогда, когда $Y \subseteq X''$. Заметим, что данный критерий тривиальным образом выполняется для $X \Rightarrow X''$.

Из определения замыкания и формулы (3) следует, что для всякого контекста $K = (G, M, I)$ и любого множества $X \subseteq M$ поддержка X'' относительно $K = (G, M, I)$ совпадает с поддержкой X :

$$\delta(X'') = \delta(X). \quad (8)$$

Таким образом, если $X \subseteq M$ является частым множеством в $K = (G, M, I)$, то X'' также является частым в этом контексте. Следует отметить, что в худшем случае число частых замкнутых множеств признаков контекста $K = (G, M, I)$ совпадает с числом частых множеств признаков и экспоненциально зависит от $|M|$. Однако на практике обычно число частых замкнутых множеств значительно меньше числа частых множеств. Примеры контекстов с полиномиальным относительно $|M|$ числом частых замкнутых множеств можно найти в [16]. Использование формулы (8) позволяет при поиске ассоциативных правил вместо частых множеств признаков применять частые замкнутые множества и тем самым сокращать пространство поиска ассоциативных правил. В силу (3) и (8) расширение исходного контекста путем добавления в него новых объектов не изменяет замыкания $X'' \subseteq M$, но может расширить состав объектов, которым присущи все признаки из X'' . Это означает, что при такой трансформации исходного контекста возможно лишь увеличение поддержек для ранее найденных замкнутых множеств признаков. Поэтому, если эти множества были частыми, они по-прежнему остаются частыми.

Задача поиска ассоциативных правил и известные алгоритмы ее решения

Пусть заданы контекст $K = (G, M, I)$ и δ_0, γ_0 – вещественные числа из $[0, 1]$. Будем говорить, что $X \Rightarrow Y$ является (δ_0, γ_0) -ассоциативным правилом в $K = (G, M, I)$, если выполняются два условия:

$$\gamma_0 \leq \gamma(X \Rightarrow Y) \leq 1, \quad (9)$$

$$\delta_0 \leq \delta(X \Rightarrow Y) \leq 1. \quad (10)$$

Требуется найти для заданного контекста $K = (G, M, I)$ множество AR всех (δ_0, γ_0) -ассоциативных правил. Заметим, что искомым набор правил AR параметризован относительно пороговых значений δ_0 и γ_0 . Например, при $\delta_0 = 0$ условие (10) отражает естественные границы поддержки. Данная ситуация свидетельствует о том, что нет ограничений на частоту появления признаков $X \cup Y$ в $K = (G, M, I)$. При $\gamma_0 = 1$ условие (9) приводит к равенству $\gamma(X \Rightarrow Y) = 1$. В этом случае будем иметь строгие ассоциативные правила.

Решение поставленной задачи предполагает выполнение следующих двух этапов: вначале построение различных подмножеств множества M , вычисление для каждого из них поддержки и проверка условия (10); далее генерация ассоциативных правил с учетом найденных частых наборов и проверка условия (9). Наличие на обоих этапах комбинаторных переборов приводит к значительным вычислительным затратам для нахождения решения данной задачи. Очевидно, что, чем больше пороговые значения, тем быстрее находится соответствующий набор ассоциативных правил и тем меньшее число правил будет содержать этот набор. К сожалению, в худшем случае время поиска всех (δ_0, γ_0) -ассоциативных правил в заданном контексте $K = (G, M, I)$ экспоненциально зависит от $|M|$.

К настоящему времени разработано большое число алгоритмов поиска ассоциативных правил. Их обзор представлен в работах [15, 16]. Основополагающими среди них являются алгоритмы Apriori и Close. Алгоритм Apriori использует свойство антимонотонности функции поддержки [3, 9]. Он генерирует (δ_0, γ_0) -ассоциативные правила для любых δ_0 и γ_0 исходя из найденных частых множеств признаков. Алгоритм Close представляет класс алгоритмов, генерирующих только $(\delta_0, 1)$ -ассоциативные правила, то есть строгие ассоциативные правила. Этот алгоритм порождает строгие ассоциативные правила из частых замкнутых наборов признаков [15]. Переход от частых множеств к частым замкнутым множествам позволяет сузить пространство поиска. Другим достоинством алгоритма Close и его многочисленных версий является способность формировать минимаксные ассоциативные правила.

Пример 1. Рассмотрим контекст $K = (G, M, I)$, представленный в таблицах 1 и 2, где $G = \{g_1, g_2, g_3, g_4, g_5\}$ – множество объектов, $M = \{a, b, c, d, e\}$ – множество признаков, I – матрица инцидентности. Заметим, что именно этот контекст традиционно используется в публикациях по алгоритмам поиска ассоциативных правил для демонстрации и тестирования этих алгоритмов [9]. В таблицах 1, 2 и далее при написании множеств для краткости опущены фигурные скобки, запятые между элемен-

тами этих множеств и элементы расположены в лексикографическом порядке. Например, вместо $\{a, b, c\}$ записано abc , а вместо $\{a, b\} \Rightarrow \{c\} - ab \Rightarrow c$.

Таблица 1

Исходный контекст

Table 1

Input context

Объект	Признаки, присущие объектам
g_1	acd
g_2	bce
g_3	$abce$
g_4	be
g_5	$abce$

Таблица 2

Матрица инцидентности исходного контекста

Table 2

The incidence matrix of the input context

	a	b	c	d	e
g_1	1	0	1	1	0
g_2	0	1	1	0	1
g_3	1	1	1	0	1
g_4	0	1	0	0	1
g_5	1	1	1	0	1

Для рассматриваемого контекста при $\delta_0 = 1/5$ и $\gamma_0 = 1$ алгоритм Apriori приводит к семнадцати строгим ассоциативным правилам:

$AR = \{a \Rightarrow c, d \Rightarrow a, e \Rightarrow b, b \Rightarrow e, d \Rightarrow c, ab \Rightarrow c, ae \Rightarrow b, ab \Rightarrow e, cd \Rightarrow a, ad \Rightarrow c, d \Rightarrow ac, ae \Rightarrow c, ce \Rightarrow b, bc \Rightarrow e, ace \Rightarrow b, abe \Rightarrow c, abc \Rightarrow e\}$. (11)

Полученное множество AR содержит много избыточных правил в том смысле, что, если их исключить, то они выводимы из оставшихся правил. Это следующие ассоциативные правила: $\{d \Rightarrow c, ab \Rightarrow c, ae \Rightarrow b, ab \Rightarrow e, cd \Rightarrow a, ad \Rightarrow c, d \Rightarrow ac, ae \Rightarrow c, ce \Rightarrow b, bc \Rightarrow e, ace \Rightarrow b, abe \Rightarrow c, abc \Rightarrow e\}$.

Некоторые из них удаляются алгоритмом Close в процессе построения множества AR . В результате алгоритм Close извлекает лишь восемь минимаксных строгих ассоциативных правил:

$AR = \{a \Rightarrow c, b \Rightarrow e, e \Rightarrow b, d \Rightarrow ac, bc \Rightarrow e, ce \Rightarrow b, ab \Rightarrow ce, ae \Rightarrow bc\}$, (12)

что в два раза меньше чем в (11). Алгоритм Close находит минимаксные строгие ассоциативные правила по следующим частым замкнутым наборам признаков: $c, ac, be, bce, acd, abce$. Все эти правила имеют допустимые поддержки: $\delta(a \Rightarrow c) = 3/5$, $\delta(b \Rightarrow e) = 4/5$, $\delta(e \Rightarrow b) = 4/5$, $\delta(d \Rightarrow ac) = 1/5$, $\delta(bc \Rightarrow e) = 3/5$, $\delta(ce \Rightarrow b) = 3/5$, $\delta(ab \Rightarrow ce) = 2/5$, $\delta(ae \Rightarrow bc) = 2/5$.

Классический алгоритм Close

Суть классического алгоритма Close заключается в пошаговом извлечении генераторов и частных

замкнутых наборов признаков [11, 15]. Множество $\rho \subseteq M$ называется генератором замкнутого набора признаков $X \subseteq M$, $X = X''$, тогда и только тогда, когда $\rho'' = X$ и не существует другого множества $\tau \subseteq M$, такого, что $\tau \subset \rho$ и $\tau'' = X$. Другими словами, генератор замкнутого набора признаков X – наименьшее по мощности множество признаков, имеющее замыкание X . Число признаков, входящих в генератор ρ , называется мощностью этого генератора. Если $|\rho| = k$, то ρ является k -генератором.

На вход алгоритма Close подаются исходный контекст $K = (G, M, I)$ и пороговое значение δ_0 . Изначально множество AR считается пустым и $k = 1$. На первом шаге в качестве k -генераторов рассматриваются все одноэлементные подмножества множества M . Замыкание ρ_k'' для генератора ρ_k вычисляется по формуле (3). Поддержка для ρ_k'' находится по формуле (4). Если $\delta(\rho_k'') \geq \delta_0$, то по частому замкнутому множеству ρ_k'' строится минимаксное строгое ассоциативное правило

$$\rho_k \Rightarrow \rho_k'' \setminus \rho_k \quad (13)$$

и сохраняется в AR . Согласно (6) и (8), для него $\delta(\rho_k \Rightarrow \rho_k'' \setminus \rho_k) = \delta(\rho_k'') \geq \delta_0$, $\gamma(\rho_k \Rightarrow \rho_k'' \setminus \rho_k) = 1$. То, что ассоциативное правило (13) является минимаксным, следует из определения генератора. Очевидно, что различные генераторы могут порождать одинаковые замыкания. Так, для контекста из примера 1 имеем: $b'' = be$, $e'' = be$. Поэтому (12) содержит минимаксные строгие ассоциативные правила $b \Rightarrow e$ и $e \Rightarrow b$.

После генерации ассоциативного правила по ρ_k'' создаются кандидаты в $(k+1)$ -генераторы для следующей итерации. Каждый такой кандидат формируется путем объединения двух k -генераторов, обладающих одинаковыми первыми $k-1$ признаками. Далее проверяется, вложен ли найденный кандидат в ρ_k'' . Если вложен, то он исключается из рассмотрения. После нахождения всех $(k+1)$ -генераторов осуществляется переход к следующей итерации. Алгоритм Close завершает работу, когда исчерпаны все генераторы.

Множество ассоциативных правил, полученных в результате работы алгоритма Close, образует минимаксный базис строгих ассоциативных правил контекста $K = (G, M, I)$. Корректность алгоритма Close доказана в работе [15]. Алгоритм Close может быть модифицирован так, чтобы в результирующее множество AR не попадали заведомо избыточные строгие ассоциативные правила.

Алгоритм MClose для построения сжатого строгого базиса

Верна следующая теорема.

Для любого контекста $K = (G, M, I)$ и произвольных $X, Y, Z, W \subseteq M$ справедливы следующие свойства строгих ассоциативных правил:

D_1 . Рефлексивность: $X \Rightarrow X$.

D_2 . Пополнение посылки:

если $X \Rightarrow Y$, то $X \cup Z \Rightarrow Y$.

D_3 . Аддитивность:

если $X \Rightarrow Y$ и $X \Rightarrow Z$, то $X \Rightarrow Y \cup Z$.

D_4 . Проективность:

если $X \Rightarrow Y$ и $Z \subseteq Y$, то $X \Rightarrow Z$.

D_5 . Транзитивность:

если $X \Rightarrow Y$ и $Y \Rightarrow W$, то $X \Rightarrow W$.

D_6 . Псевдотранзитивность:

если $X \Rightarrow Y$ и $Y \cup Z \Rightarrow W$, то $X \cup Z \Rightarrow W$.

Свойства (или выводимости) D_1 – D_6 позволяют из некоторого множества строгих ассоциативных правил вывести многие другие правила без дополнительного сканирования контекста. С одной стороны, именно выводимости D_1 – D_6 обуславливают экспоненциальное число возможных строгих ассоциативных правил для рассматриваемого контекста и неоднозначность их представления. С другой стороны, они предоставляют возможность построить для множества строгих ассоциативных правил сжатый строгий базис (*concise strong basis*), компактно описывающий связи между данными изучаемой предметной области.

Примечательно, что выводимости, подобные D_1 – D_6 , справедливы для функциональных зависимостей, имеющих место в теории реляционных БД, где их принято называть аксиомами Амстронга. Такое совпадение не случайно, поскольку строгие ассоциативные правила можно рассматривать как особый случай функциональных зависимостей [5]. В работе [12] выводимости D_1 – D_6 доказаны применительно к функциональным зависимостям. Однако они могут быть доказаны непосредственно для ассоциативных правил на основе соответствий Галуа. Важно, что выводимости D_1 , D_3 , D_4 , D_5 гарантируют сохранение поддержки. Это означает, что результатом применения их $(\delta_0, 1)$ -ассоциативным правилом всегда является $(\delta_0, 1)$ -ассоциативное правило. Поэтому эти выводимости применяются в алгоритме MClose для построения сжатого строгого базиса. Введем определения избыточного ассоциативного правила и сжатого строгого базиса для множества $(\delta_0, 1)$ -ассоциативных правил.

Пусть AR – множество строгих ассоциативных правил, каждое из которых справедливо для контекста $K = (G, M, I)$. Будем говорить, что строгое ассоциативное правило $X \Rightarrow Y$ логически следует из множества AR , если оно может быть выведено из AR с помощью выводимостей D_1 , D_3 , D_4 , D_5 . Тот факт, что строгое ассоциативное правило $X \Rightarrow Y$ логически следует из AR , будем обозначать так: $AR \vdash X \Rightarrow Y$.

Строгое ассоциативное правило $X \Rightarrow Y$ называется избыточным в AR , если $AR \setminus \{X \Rightarrow Y\} \vdash X \Rightarrow Y$. Множество строгих ассоциативных правил неизбыточное, если оно не содержит избыточных стро-

гих ассоциативных правил. Множество CSB называется сжатым строгим базисом множества AR , если оно неизбыточное и состоит только из минимаксных строгих ассоциативных правил. Данное определение сжатого строгого базиса указывает способ его нахождения – генерация минимаксных строгих ассоциативных правил (например, с помощью алгоритма Close) и устранение среди них избыточных.

Распознавание избыточного строгого ассоциативного правила в AR основано на проверке логического следования

$$AR \setminus \{X \Rightarrow Y\} \not\vdash X \Rightarrow Y. \quad (14)$$

Алгоритм такой проверки использует понятие замыкания множества признаков относительно множества AR и является полиномиальным относительно $|M|$ и $|AR|$. Замыканием множества $X \subseteq M$ относительно AR (обозначается X^+) называется множество всех признаков $m \in M$, таких, что верно логическое следование $AR \vdash X \Rightarrow m$. Заметим, что неизменно $X^+ \subseteq M$. Из выводимостей D_1 , D_3 , D_4 вытекает справедливость следующего утверждения.

Утверждение 3. Логическое следование $AR \vdash X \Rightarrow Y$ имеет место тогда и только тогда, когда $Y \subseteq X^+$.

Отсюда всегда $AR \vdash X \Rightarrow X^+$, $AR \not\vdash X \Rightarrow X^+ \setminus X$. В силу утверждения 3, чтобы убедиться в справедливости (14), достаточно вычислить X^+ относительно $AR \setminus \{X \Rightarrow Y\}$ и проверить включение $Y \subseteq X^+$. Если $Y \subseteq X^+$, то строгое ассоциативное правило $X \Rightarrow Y$ избыточно в AR , иначе – не избыточно.

Алгоритм вычисления X^+ целиком базируется на выводимостях D_1 , D_3 , D_4 , D_5 и сводится к выполнению следующих действий. Сначала полагается $X^+ = X$. Далее осуществляются просмотр правил из AR и пополнение замыкания по следующему принципу: если для правила $Y \Rightarrow Z \in AR$ верно включение $Y \subseteq X^+$, то множество Z добавляется к X^+ . Этот процесс повторяется до тех пор, пока изменяется X^+ . Поскольку множества M и AR конечные, то процесс вычисления X^+ конечен.

Заметим, что процесс исключения избыточных строгих ассоциативных правил не требует доступа к контексту $K = (G, M, I)$ и поэтому время его выполнения незначительно по сравнению со временем получения частых замкнутых множеств признаков. Чтобы исключить добавление заведомо избыточного строгого ассоциативного правила в AR , необходимо всякий раз после построения ρ_k'' выполнять следующие действия. Если посылка ρ_k найденного правила не равна ρ_k'' , то необходимо найти замыкание ρ_k^+ относительно вычисленного множества AR . Если $\rho_k^+ = \rho_k''$, то минимаксное ассоциативное правило $\rho_k \Rightarrow \rho_k'' / \rho_k$ является избыточным (по утверждению 3), иначе оно включается в AR . После завершения генерации минимаксных строгих ассоциативных правил обязательно требу-

ется дополнительный просмотр результирующего множества AR с целью обнаружения оставшихся избыточных правил. Такие правила вполне возможны: они избыточные по отношению к ранее выявленным правилам, однако после пополнения AR новыми правилами могут оказаться избыточными. Таким образом, по построению результирующее множество AR состоит из минимаксных строгих ассоциативных правил и является избыточным. Заметим, что оперативное удаление избыточных правил сдерживает рост мощности AR и снижает время выполнения алгоритма. Опишем алгоритм MClose.

Алгоритм 1. MClose

```

1: begin
2:    $AR \leftarrow \emptyset$ 
3:    $k \leftarrow 1$ 
4:   while  $\rho_k \neq \emptyset$ 
5:     Gen-Closure ( $\rho_k$ )
6:     if  $\delta(\rho_k) \geq \delta_0$ 
7:       if  $\rho_k \neq \rho_k''$ 
8:          $\rho_k^+ \leftarrow SX(\rho_k)$ 
9:       end if
10:      if  $\rho_k^+ \neq \rho_k''$ 
11:         $AR \leftarrow (\rho_k \Rightarrow \rho_k'' \setminus \rho_k)$ 
12:      end if
13:    end if
14:    Gen-Generator ( $k + 1$ )
15:     $k \leftarrow k + 1$ 
16:  end while
17:  Non-Redundancy ( $AR$ )
18: end

```

В описании алгоритма MClose процедуры Gen-Closure и Gen-Generator аналогичны одноименным процедурам классического алгоритма Close (см. [11, 15]). Процедура SX осуществляет построение замыкания ρ_k^+ . Процедура Non-Redundancy, выполняемая на шаге 17 алгоритма MClose, устраняет в AR избыточные строгие ассоциативные правила после завершения его генерации.

Пример 2. Для контекста, представленного в таблицах 1 и 2, при $\delta_0 = 1/5$ и $\gamma_0 = 1$ минимаксный базис состоит из восьми правил (12), в котором избыточными являются

$$bc \Rightarrow e, ce \Rightarrow b, ab \Rightarrow ce, ae \Rightarrow bc. \quad (15)$$

Сжатый строгий базис, построенный алгоритмом MClose, содержит только четыре минимаксных строгих ассоциативных правила:

$$CSB = \{a \Rightarrow c, b \Rightarrow e, e \Rightarrow b, d \Rightarrow ac\}.$$

Заметим, что на основе CSB можно получить всякое правило из (15) с помощью алгоритма вычисления X^+ . Так, чтобы найти максимальное следствие для посылки bc , достаточно вычислить bc^+ относительно CSB . В результате имеем $bc^+ = bce$. Отсюда $CSB \models bc \Rightarrow e$. Поскольку $bc^+ = bc'' = bce$ и $e \in bc''$, то ассоциативное правило $bc \Rightarrow e$ является строгим.

Программная реализация алгоритма MClose и результаты экспериментов

Алгоритм MClose программно реализован на языке программирования C++ в среде разработки Embarcadero RAD Studio XE8. Исходными данными для него служат контекст $K = (G, M, I)$ и пороговое значение поддержки. Для ввода исходных данных имеется интерфейс (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/7.jpg). Для удобства пользователя пороговое значение вводится в виде целого положительного числа $\delta_0 \cdot |G|$. Результатом работы алгоритма MClose являются все выявленные частые замкнутые множества признаков и сжатый строгий базис (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/8.jpg). Значения поддержек масштабируются и выводятся в виде целых положительных чисел.

В программе существует возможность изменения результирующего множества ассоциативных правил путем добавления и удаления отдельных правил. Это сделано для того, чтобы эксперт мог исключить из рассмотрения ассоциативные правила, которые, на его взгляд, не отвечают реальной действительности, и добавить существенные правила. Допускается сокращение признакового пространства (вкладка «Анализ признаков») с помощью методов, описанных в работе [17]. Предусмотрена функция вычисления следствия для всякой заданной посылки относительно сжатого строгого базиса. Для оценки эффективности алгоритма MClose были также программно реализованы алгоритмы Apriori и Close.

Алгоритмы Apriori, Close и MClose сравнивались по числу сгенерированных строгих ассоциативных правил и времени работы. Эксперименты осуществлялись на компьютере с процессором Intel® Core™ i5 CPU & 2.30 GHz и ОЗУ размером 4 Гб. Эксперименты выполнялись на контекстах различной размерности и плотности, сгенерированных случайным образом. Результаты экспериментов представлены в таблице 3.

Таблица 3

Результаты экспериментов

Table 3

Results of experiments

Характеристика контекста $K = (G, M, I)$			Алгоритм					
			Apriori		Close		MClose	
$ G $	$ M $	σ	1	2	1	2	1	2
20	10	0,38	1797	17562	45	250	22	297
30	10	0,39	2029	18347	46	374	19	412
30	10	0,55	15438/	187202	69	390	20	484
50	10	0,53	27769	375178	46	78	13	124
500	10	0,53	27769	376154	42	124	13	168
10000	10	0,53	27769	378400	42	671	13	872

Примечание: в колонках под цифрой 1 – число извлеченных строгих ассоциативных правил, под цифрой 2 – время, мс.

В таблице 3 для всякого анализируемого контекста $K = (G, M, I)$ указаны $|G|$ – число объектов, $|M|$ – число признаков, $\sigma = n / (|G| \cdot |M|)$ – плотность контекста, где n задает число единичных элементов матрицы инцидентности I . Контекст из 10 000 объектов был сформирован многократным копированием контекста, состоящего из 500 объектов.

Из таблицы 3 видно, что алгоритмы Close и MClose эффективнее алгоритма Apriori как по числу извлеченных строгих ассоциативных правил, так и по времени работы. Алгоритм MClose по времени работы сопоставим с алгоритмом Close. Однако алгоритм MClose позволяет более чем в два раза уменьшить мощность минимаксного базиса, формируемого алгоритмом Close.

Заключение

Современные методы и средства поиска ассоциативных правил в больших массивах данных приводят к значительному числу правил, большинство из которых являются избыточными. Усилия многих исследователей направлены на разработку методов устранения избыточности в представлении ассоциативных правил. Для решения этой проблемы применительно к строгим ассоциативным правилам предложен алгоритм MClose, являющийся модификацией известного алгоритма Close и основанный на свойствах частых замкнутых множеств. Предложенный алгоритм формирует для заданного контекста сжатый строгий базис – не избыточное множество минимаксных строгих ассоциативных правил с сохранением поддержки. Вычислительные эксперименты показали, что алгоритм MClose существенно сокращает число извлеченных строгих ассоциативных правил без потери информации о связях между данными анализируемой предметной области. Перспективы исследования, направленные на устранение избыточности для ассоциативных правил с любыми поддержками и достоверностями.

Литература

1. Батура Т.В. Модели и методы анализа компьютерных социальных сетей // Программные продукты и системы. 2013. № 3. С. 130–137.
2. Платонов В.В., Семенов П.О. Методы сокращения размерности в системах обнаружения сетевых атак // Проблемы информационной безопасности. Компьютерные системы. 2012. № 3. С. 40–45.
3. Ilayaraja M., Meyyappan T. Mining medical data to identify frequent diseases using Apriori algorithm. Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, 2013, pp. 194–199.
4. Городецкий В.И., Самойлов В.В. Ассоциативный и причинный анализ и ассоциативные байесовские сети // Тр. СПИИРАН. 2009. Вып. 9. С. 13–65.
5. Кузнецов С.О. Автоматическое обучение на основе анализа формальных понятий // Автоматика и телемеханика. 2001. № 10. С. 3–27.
6. Биркгоф Г., Барти Т. Современная прикладная алгебра. СПб: Лань, 2005. 400 с.
7. Гуров С.И. Булевы алгебры, упорядоченные множества, решетки: определения, свойства, примеры. М.: ЛИБРОКОМ, 2013. 352 с.
8. Ganter B., Wille R. Formal Concept Analyses: mathematical foundations, Springer Science and Business Media, 2012, 314 p.
9. Zhang C., Zhang S. Association rules mining. Springer, 2002, 240 p.
10. Geng L., Hamilton H.J. Interestingness measures for data mining: a survey. ACM Computing Surveys, 2006, vol. 38, no. 3, article 9.
11. Pasquier N., Bastide Y., Taouil R. & Lakhal L. Generating a condensed representation for association rules. Jour. of Intelligent Information Systems, 2005, vol. 24, no. 1, pp. 29–60.
12. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987. 608 с.
13. Duquenne V., Obiedkov S.A. Attribute-incremental construction of the canonical implication basis. Annals of Mathematics and Artificial Intelligence, 2007, vol. 49, no. 1–4, pp. 77–99.
14. Rudolph S. Some notes on pseudo-closed sets. Proc. ICFA 2007, LNCS, Springer, 2007, vol. 4390, pp. 151–165.
15. Zaki M.J., Hsiao Ch.-J. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. IEEE Transaction on Knowledge and Data Engineering, 2005, vol. 17, no. 4, pp. 462–478.
16. Uno T., Asai T., Uchida Y., Arimura H. An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases. Proc. DS'04, LNAI 3245, 2004, pp. 16–31.
17. Быкова В.В., Катаева А.В. Методы и средства анализа информативности признаков при обработке медицинских данных // Программные продукты и системы. 2016. № 2. С. 172–178.

A CONTRACTED REPRESENTATION OF STRONG ASSOCIATIVE RULES IN DATA ANALYSIS

V.V. Bykova¹, Dr. Sc. (Physics and Mathematics), Professor, bykvalen@mail.ru

A.V. Kataeva¹, Postgraduate Student, kataeva_av@mail.ru

¹ Siberian Federal University, Svobodny Ave. 79, Krasnoyarsk, 660041, Russian Federation

Abstract. Modern methods and means of searching for association rules in big data lead to a significant number of rules, many of which are redundant. Redundant association rules are generally of no value, but they can misinform. To solve this problem, the paper proposes an algorithm MClose, which is a modification of the algorithm Close.

It is known that Close algorithm might help to construct mini-max basis for strict association rules (association rules with the confidence of 1). Mini-max basis consists of only min-max association rules. Association rules with minimal antecedent and maximal consequent are called min-max association rules. Such rules are interesting for experts. However, mini-max basis may contain redundant association rules. The algorithm MClose immediately eliminates redundant association rules when creating mini-max basis. The resulting basis is called concise strong basis (CSB). Redundant association rules might always be obtained from the CSB without sacrificing their support and confidence, without references to the data set. Algorithm MClose is based on Galois connection. MClose algorithm is also based on derivability, which are similar on Armstrong axioms for functional dependencies.

Experiments have shown that running time of algorithm MClose is comparable with the algorithm Close. However, it reduces the number of association rules mini-max basis about twice. We provide a description of the program which presents MClose and Close algorithms.

Keywords: data mining, Galois connection, closed sets, association rules, non-redundancy, concise strong basis.

References

1. Batura T.V. Social networks analysis models and methods. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3, pp. 130–137 (in Russ.).
2. Platonov V.V., Semenov P.O. Dimension reduction in network attacks detection systems. *Problemy informatsionnoy bezopasnosti. Kompyuternye sistemy* [Information Security Problems. Computer Systems]. 2012, no. 3, pp. 40–45 (in Russ.).
3. Ilayaraja M., Meyyappan T. Mining medical data to identify frequent diseases using Apriori algorithm. *Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE*, 2013, pp. 194–199.
4. Gorodetsky V.I., Samoylov V.V. Association and casual rule mining using associative bayesian networks. *Tr. SPIIRAN* [SPIIRAS Proc.]. 2009, no. 9, pp. 13–65 (in Russ.).
5. Kuznetsov S.O. Automatic learning using analysis of formal concepts. *Avtomatika i telemekhanika* [Automation and Telemechanics]. 2001, no. 10, pp. 3–27 (in Russ.).
6. Birkhoff G., Barti T. *Sovremennaya prikladnaya algebra* [Modern Applied Algebra]. St. Petersburg, Lan Publ., 2005, 400 p.
7. Gurov S.I. *Bulevy algebrы, uporyadochennyye mnozhestva, reshetki: opredeleniya, svoystva, primery* [Boolean algebra, ordered sets, grids: definitions, features, examples]. Moscow, Librokom Publ., 2013, 352 p.
8. Ganter B., Wille R. *Formal Concept Analyses: mathematical foundations*. Springer Science and Business Media Publ., 2012, 314 p.
9. Zhang C., Zhang S. *Association Rules Mining*. Springer Publ., 2002, 240 p.
10. Geng L., Hamilton H.J. Interestingness measures for data mining: a survey. *ACM Computing Surveys*. 2006, vol. 38, no. 3, article 9.
11. Pasquier N., Bastide Y., Taouil R., Lakhal L. Generating a condensed representation for association rules. *Jour. of Intelligent Information Systems*. 2005, vol. 24, no. 1, pp. 29–60.
12. Meyer D. *Teoriya relyatsionnykh baz dannykh* [The Relational Database Theory]. Moscow, Mir Publ., 1987, 608 p.
13. Duquenne V., Obiedkov S.A. Attribute-incremental construction of the canonical implication basis. *Annals of Mathematics and Artificial Intelligence*. 2007, no. 49 (1–4), pp. 77–99.
14. Rudolph S. Some notes on pseudo-closed sets. *Proc. ICFCA 2007*. LNCS, Springer Publ., 2007, vol. 4390, pp. 151–165.
15. Zaki M.J., Hsiao Ch.-J. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. *IEEE Trans. on Knowledge and Data Engineering*. 2005, vol. 17, no. 4, pp. 462–478.
16. Uno T., Asai T., Uchida Y., Arimura H. An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases. *Proc. DS'04*. LNAI Publ., 2004, no. 3245, pp. 16–31.
17. Bykova V.V., Katayeva A.V. Methods and tools for analysing informative features when processing medical data. *Programmnye produkty i sistemy* [Software & Systems]. 2016, no. 2 (114), pp. 172–178 (in Russ.).

Примеры библиографического описания статьи

1. Быкова В.В., Катаева А.В. Сжатое представление строгих ассоциативных правил в анализе данных // Программные продукты и системы. 2017. Т. 30. № 2. С. 187–195; DOI: 10.15827/0236-235X.118.187-195.
2. Bykova V.V., Kataeva A.V. A contracted representation of strong associative rules in data analysis. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 187–195 (in Russ.); DOI: 10.15827/0236-235X.118.187-195.

EFFECTIVE ALGORITHM FOR CONSTRUCTING ASSOCIATIVE RULES

V.A. Billig¹, Ph.D. (Engineering), Senior Researcher, Associate Professor, Vladimir-Billig@yandex.ru

¹ Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation

Constructing associative rules is one of the most important algorithms for extracting knowledge from databases. All modern algorithms are somehow connected with Apriori algorithm proposed in R. Agrawal's and his co-authors' works published more than 20 years ago and now considered classical. The known effective implementations of the algorithm are connected with database compression and presentation of data structure as a tree, which allows effective evaluation of support and other characteristics of associative rules.

The proposed ConApriori algorithm does not use the above idea. We regard database transactions as enumeration given by a scale. This allows instant calculation of the basic algorithm operation determining whether some set is a subset of another set or not. The calculations are reduced to several logical computer commands. Enumeration also allows us to treat the transaction in the internal presentation as a number, preserving the meaning of the transaction elements in their external presentation at the same time.

Another idea used in the algorithm allows us to construct most of the confident rules on the basis of those previously built.

This article provides evidence for correctness of the algorithm as well as evaluate its complexity. We analyze its effectiveness compared with other known algorithm implementations. Possibility of parallelization is also considered.

Keywords: data mining, apriori, associative rules, support, confidence, lift, enumeration, scale, database, transaction, knowledge discovery, parallel computation, correctness, complexity, anti-monotony.

With the exponential growth of the stored data, data mining algorithms start playing a great role. Apriori algorithm proposed by R. Agrawal and his co-authors [1–3] is particularly important. The algorithm is based on the antimonotony property of the function calculating the item set frequency. As a result, the set of k -length frequent item sets can be built using $(k-1)$ -length frequent item sets. This allows avoiding an exhaustive search and constructing frequent item sets and associative rules within reasonable time. The idea is used in all algorithms of constructing associative rules.

An important contribution to constructing effective algorithms was made by F. Bodon [4]. He stated that the algorithm's effectiveness is defined by three factors: the means of generating candidates to frequent sets, the structure of data presentation and implementation details. Bodon proposed a tree structure for transaction presentation, showed the structure's effectiveness as compared with other means of data compression.

One of the modern and probably most effective algorithms is the implementation by Ch. Borgelt [5–7] who has been developing it for more than 20 years. In this algorithm, the database is compressed to a transaction tree. The current version of the algorithm is free at his site [8].

Frequent set and associative rule algorithms have a wide application range, thus both the algorithms and the tasks taking into consideration the specific character of rule application are diverse. We will mention only the works closely correlated with the algorithm considered in the present article.

The first work on building multilevel associative rules was that of J. Han and Y. Fu [9]. In [10] the authors analyze effective building of multilevel rules. In the summarizing article by Ch. Borgelt [11] we find a

description of several algorithms. The fundamental book by J. Han, M. Kamber and J. Pei [12] devoted to the conceptions and methods of knowledge extraction from database considers various algorithms and approaches to the construction of associative rules in the broad aspect of overall approach to data mining.

The present article is organized as follows. Section 2 introduces the problem. Section 3 describes the basic conceptions of *ConApriori* algorithm and proves its correctness. Section 4 gives evaluation of its complexity. Section 5 is devoted to possible optimizations of the algorithm. Section 6 considers some constraints. Section 7 describes some details of implementation. Section 8 is devoted to approbation of the algorithm and its effectiveness compared with Ch. Borgelt's algorithm realization. Section 9 draws a conclusion.

Problem formulation

Consider database db with a set of objects:

$$db = \{v_1, v_2, \dots, v_N\}. \quad (1)$$

Each object v_k has a certain set of properties P_k :

$$P_k = \{q_{k1}, q_{k2}, \dots, q_{km}\}. \quad (2)$$

The sets P_k are subsets of the set of *Properties* common to all objects:

$$P_k \subseteq Properties = \{q_1, q_2, \dots, q_M\}. \quad (3)$$

We assume that all properties of the objects are binary, i.e., object v_k can either have or not have the property of q_j . Further, we will show how this constraint can be reduced.

Next, we will consider dbp database containing the object properties since we are not interested in the objects themselves:

$$dbp = \{P_1, P_2, \dots, P_N\}. \quad (4)$$

Let X be an item set of properties:

$$X = \{q_{i1}, q_{i2}, \dots, q_{ir}\}. \quad (5)$$

Frequency of the item set or its support in database *dbp* will be defined as a proportion of the records of the database containing the set *X*:

$$Support(X) = \frac{|P_k : X \subseteq P_k|}{N}. \quad (6)$$

An associative rule has the form:

$$X \rightarrow Y. \quad (7)$$

Here *X* and *Y* are not empty and disjoint sets of properties:

$$X \cap Y = \emptyset. \quad (8)$$

An important characteristic of the rule is its frequency (support):

$$Support(Rule) = Support(X \rightarrow Y) = Support(X \cup Y). \quad (9)$$

A rule should be considered as an implication, the “if-then” rule. If *X* is in a certain set, then so is *Y*. In other words, appearance of *X* implies occurrence of *Y*. For the rule to be of practical value it should have a high degree of confidence.

Confidence is the second important characteristic of the rule. We will define it as proportion of the records of the database that contain *X* and *Y* among the records containing *X*:

$$Confidence(Rule) = Confidence(X \rightarrow Y) = \frac{|P_k : X \cup Y \subseteq P_k|}{|P_k : X \subseteq P_k|}. \quad (10)$$

It is easy to see that confidence can be expressed through the *Support* function:

$$Confidence(Rule) = Confidence(X \rightarrow Y) = \frac{Support(Rule)}{Support(X)}. \quad (11)$$

We define the length of the rule *L(Rule)* as the sum of the pair (*m*, *n*), where *m* and *n* are lengths of sets *X* and *Y*, respectively.

The goal of the algorithm. We wish to find all *Associative Rules* with support and confidence more than specified minimum values:

$$Rules = \{Rule | Support(Rule) > support_min \& Confidence(Rule) > Confidence_min\}. \quad (12)$$

The basic idea of Apriori algorithm. We call the set *X* frequent if *Support(X) > support_min*.

The following statement is true:

If X is a frequent set, all its subsets are also frequent sets.

A contrapositive statement is more important:

If X is not a frequent set, all its supersets are not frequent sets either.

This property called Antimonotony allows excluding a large number of sets from our consideration. Construction of all frequent rules of length *m* can be performed by extending the set of frequent rules of length *m-1*. Then in the built set of frequent rules we can choose the rules with the required confidence.

The lift parameter. In addition to frequency and confidence, we consider other characteristics of the rules.

An important additional feature of the rules is the *lift* parameter specifying the correlation between two events, which are antecedent of the rule and its consequent.

Let us discuss the role of the parameter. We will consider *Support* parameter specifying the frequency of set *X* in the database as the probability of event *X*.

Then for the rule $X \rightarrow Y$ and the symmetric rule $Y \rightarrow X$ the *lift* parameter will be defined as follows:

$$\begin{aligned} lift &= \frac{P(X, Y)}{P(X) * P(Y)} = \frac{P(X) * P(Y/X)}{P(X) * P(Y)} = \\ &= \frac{P(Y) * P(X/Y)}{P(X) * P(Y)} = \frac{P(Y/X)}{P(Y)} = \frac{P(X/Y)}{P(X)}. \end{aligned} \quad (13)$$

For independent events, probability $P(X, Y)$ of co-occurrence of events *X* and *Y* is the product of probabilities of occurrence of the events $P(X) * P(Y)$, so that the *lift* parameter in this case is equal to 1. For dependent events the conditional probability $P(Y/X)$ (a probability of occurrence of event *Y* on condition that event *X* takes place) can be higher than probability $P(Y)$. By analogy, the conditional probability $P(X/Y)$ can be higher than probability $P(X)$. For fully dependent events, when a single event clearly implies the occurrence of another event, conditional probabilities $P(Y/X)$ and $P(X/Y)$ are equal to 1 and probabilities $P(X)$ and $P(Y)$ coincide. In this situation the *lift* parameter has a maximum value: $1/P(Y) = 1/P(X)$. For fully dependent events when a single event clearly implies impossibility of the occurrence of another event, conditional probabilities $P(Y/X)$ and $P(X/Y)$ are equal to 0 and the *lift* parameter has a minimum value equal to 0. It implies that values of the *lift* parameter are in the range $[0, \max(1/P(Y), 1/P(X))]$.

Values of *lift* close to 1 show no correlation between the premise of the rule and its conclusion, so that such rules should be considered useless, even if they are frequent and confident. Values of *lift* parameter greater than 1, approaching a maximum value, indicate a high positive correlation confirming confidence of the rule. Values of *lift* close to zero indicate a negative correlation between the premise and the conclusion. Such rules may also be useful, as well as rules with positive correlation, allowing us to make important conclusions about the existence of links between the sets of the analyzed parameters.

ConApriori algorithm. The basic ideas

There are two key ideas underlying the *ConApriori* algorithm. The first is related to the method of data presentation. The second is related to the method of constructing confident rules.

Data presentation. How to present database records containing object properties? It is possible to pre-

sent records as a list of properties in csv format (comma-separated values). In this case, the length of each record varies depending on the number of properties owned by the object. Another method (or approach) takes into account the binary nature of the properties. Each record is a string of fixed length M , where M is the total number of possible properties. The string consists of zeros and ones, where zero means absence of the corresponding property of the object and one identifies its availability. A string of zeros and ones can be viewed as a number in binary notation, so in this case, each record is presented by one integer from the range $[0, 2^M - 1]$.

The first method preserves the meaning of each record. However, its drawback is the size of a database depending not only on the number of records in the database, but also on the length of each record. The second method provides a compact presentation of the data but does not save their meaning, which is extremely important in applications, when users of a software system are experts in a relevant application area.

We have tried to avoid the difficulties. Properties of objects are defined as *enumeration*, a type well-known in programming languages. A feature of this type is that the external presentation of data is a line of text describing properties of objects. The elements of enumeration are mapped onto integers which present internal data. Moreover, to present the dataset we choose a special kind of *enumeration* type called *scale*. Scale is characterized by the fact that the k -th element of *enumeration* is displayed as the number 2^k , and the entire set can be viewed as an integer in binary notation. In the binary presentation of a number, the unit in k -th position means that the corresponding object has the k -th property in the *enumeration*.

Let us consider a simple example where three properties characterize a person: $Properties = \{good, smart, rich\}$.

In *enumeration* with the property “good” given by *scale* maps to 1, “smart” maps to 2, rich maps to 4. The set of properties $\{good, rich\}$ maps to 5 (in binary notation 101), the set $\{smart, rich\}$ maps to 6, and the rare set $\{good, smart, rich\}$ maps to 7.

Enumeration given by *scale* has an important feature. On its elements, which are numbers, we can define logical bitwise operations. This allows us to efficiently perform the basic operation, that is, to determine whether one set is a subset of another set. Let X be a set of properties and P_k be a database record. Objects X and P_k have *enumeration* type given by *scale*. Then $X \subseteq P_k$, when the following Boolean expression is true:

$$(X \& P_k) = X. \quad (14)$$

In this expression $\&$ is a bitwise conjunction running over binary presentations X and P_k , and $=$ is a comparison operation. We will use the sign $|$ for a bitwise disjunction operation. Operations $\&$ and $|$ can be regarded as operations of joining and intersecting of the corresponding sets of properties.

The chosen presentation allows the complex operation of determining the occurrence of a set X in a set Y to be performed in constant time, independent of the size of sets, almost instantly, as the implementation requires only two machine operations of the computer. Hence, the complexity of computing *Support* function is significantly simplified. Complexity of computing the function is $O(N)$, it depends linearly on the size of the database with a small constant factor.

It is necessary to mention a possible limitation inherent to this method of data presentation. While implementing the algorithm, we have a limited number of possible properties due to the finiteness of the interval presenting the integer type of the programming language.

In the following examples of recording algorithms and their fragments we use an implementation of our algorithm in the C# programming language. Note that in C# the maximum integer is 64-bit long type, and therefore the number of possible properties may not exceed 64.

Further we will show in more detail how to overcome this limitation and discuss the question of the reasonable number of properties.

Constructing confident associative rules. The basic idea of *Apriori* algorithm is associated with the iterative scheme, which allows constructing frequent rules. Frequent sets of length k are constructed on the basis of frequent sets of length $k-1$. Construction of rules does not involve infrequent sets. This is the essence of *Apriori* algorithm.

In addition to the properties of antimonotony characterizing frequent sets and rules, *ConApriori* algorithm uses a similar property characterizing confident rules. This property allows us to construct confident rules at step k on the basis of confident rules constructed at step $k-1$. The algorithm is based on the following statement:

Lemma 1.

If $X \rightarrow Y_k$ is a confident rule, any rule $X \rightarrow Y_j$, where Y_j is a proper subset of Y_k is a confident rule.

Let us prove that the statement is true.

Since Y_j is a proper subset of Y_k , the latter can be presented as: $Y_k = Y_j | add$. This record takes into account the variables specified by *enumeration*, so that the union of sets can be expressed through disjunction.

The confidence of rule $X \rightarrow Y_k$ is computed as follows:

$$Confidence(X \rightarrow Y_k) = \frac{Support(X | Y_j | add)}{Support(X)}. \quad (15)$$

The confidence of rule $X \rightarrow Y_j$ is computed as follows:

$$Confidence(X \rightarrow Y_j) = \frac{Support(X | Y_j)}{Support(X)}. \quad (16)$$

The denominators in formulas (15) and (16) coincide, and the numerator in formula (15) is less than or

equal to the numerator in formula (16) by the property of set frequency. Hence, our statement is true.

Consequence. The negation of the statement is more important than the statement itself. And here anti-monotony takes place. If rule $X \rightarrow Y_j$ is not confident, rule $X \rightarrow Y_k$ is not confident either.

It follows that confident rules with a given premise X can be constructed on the basis of the previously built confident rules with the same premise.

Note that the statement only applies to rules with a given premise. It is possible that rules $X \rightarrow Y$ and $Z \rightarrow Y$ are not confident, and rule $X \mid Z \rightarrow Y$ is a confident rule. Therefore, it is not sufficient to build confident rules at step k using only confident rules of step $k - 1$. However, it is sufficient to add confident rules, whose premise is of length k , and the conclusion is a single frequent set. We give more details about it below in considering the iterative scheme of the algorithm.

Iterative scheme of ConApriori algorithm. Consider the iterative scheme of *ConApriori* algorithm. First, at step 1 we construct a set of frequent sets of length 1 and a set of frequent confident rules of length 2 with a single premise and a single conclusion. Next, iteratively, at step k the built set of confident rules of the previous step is extended with the addition of a single frequent item in the premise or in the conclusion. To complete the construction of the set of all frequent confident rules of length $k + 1$, it is sufficient to add confident rules, their premise being taken from the set of frequent examples of length k , and their conclusion being a frequent single set. The process continues as long as it is possible to construct new rules.

Consider the process of extending a set of confident rules at each step. Suppose at step $k - 1$, there is a built frequent confident rule: $X \rightarrow Y$. Let *cand* be a frequent single set satisfying the rule. This means that *cand* is not contained in the union of X and Y of length $k - 1$ and all subsets of length $k - 1$ obtained by replacing one of the elements of the joint set X and Y by *cand* element are frequent. Checking the candidate with the given way of presenting the sets is quite easy. If checking of *cand* is successful, it is possible to generate 6 new rules:

$$\begin{array}{lll} X \mid \text{cand} \rightarrow Y; & Y \rightarrow X \mid \text{cand}; & X \rightarrow Y \mid \text{cand}; \\ Y \mid \text{cand} \rightarrow X & X \mid Y \rightarrow \text{cand}; & \text{cand} \rightarrow X/Y. \end{array} \quad (17)$$

All 6 rules have the same support – $\text{Support}(X \mid Y \mid \text{cand})$. If the support value is less than the minimum value, no new rules are built. For frequent rules, only after checking their confidence they are included in the list of confident rules at the next step. To calculate the confidence of all six rules, it is sufficient to calculate two values of the *Support* function – $\text{Support}(X \mid \text{cand})$ and $\text{Support}(Y \mid \text{cand})$.

In accordance with lemma 1, when extending confident rule $X \rightarrow Y$ we should only build confident rules of the form $X \rightarrow Y \mid \text{cand}$. However, for effectiveness other confident rules of length k are also constructed, since for five added rules it is required to additionally calculate only one value of *Support* function. A simple

check is sufficient to avoid duplication in constructing these rules. This technique does not guarantee the construction of all confident rules of length k . However, as practice shows, few rules will have to use premises not included in the set of premises of length $k - 1$ rules.

Lemma 2. *If at step $k - 1$ a set of all frequent confident rules of length k is built, the considered iterative process at step k will construct a set of all frequent confident rules of length $k + 1$.*

Indeed, the iterative process at step k builds all the rules of length $k + 1$ of the form $X \rightarrow Y \mid \text{cand}$ for each rule of length k of the previous step $X \rightarrow Y$. In accordance with lemma 1, this means that all possible confident rules will be built with the same premises as the rules of the previous step and all possible conclusions. If we add to this set all rules with all possible single conclusions and frequent premises of length k not coinciding with the premise of the previous step confident rules, the full set of confident rules of length $k + 1$ will be built.

Let us now consider the software interpretation of *ConApriori* algorithm. It is based on the *Create_Rules* method, which allows us to build confident rules. A possible implementation is as follows:

```
public void Create_Rules()
{
    iteration = 1;
    Create_Rules_1();
    bool exist_new_rules = true;
    while(exist_new_rules)
    {
        iteration++;
        exist_new_rules = Create_Rules_K();
    }
}
```

Lemma 3. *CreateRules method of ConApriori algorithm builds all frequent confident rules.*

To prove the lemma it is necessary to prove the correctness of the following Hoare triads:

$$\{Pred1\} \text{ Create_Rules_1 } \{Post1\} \quad (18)$$

$$\{Pred2\} \text{ Create_Rules_K } \{Post2\}. \quad (19)$$

Predicate *Pred1* of the precondition states that before calling *Create_Rules_1()* a database db has been built in the previously described data format. The predicate of the postcondition *Post1* states that there is a constructed set of the rules – the set of all frequent confident rules of length 1 with a single premise and a single conclusion. The truth of the triad (18) ensures that if the precondition is true, the postcondition of the method will also be true after the completion of *Create_Rules_1()* method.

Precondition *Pred2* confirms the truth of the conjunction: $\text{exist_new_rules} \ \& \ \text{Rules}(k - 1)$ – the conjunction of Boolean variable – and the statement about the construction of all confident rules of length k at step $k - 1$. Postcondition *Post2* can be written as follows:

$$(\text{Rules}(k) \ \& \ \text{exist_new_rules}) \text{ XOR } \text{!exist_new_rules}. \quad (20)$$

The truth of the triad (19) ensures that if the precondition is true, the postcondition will be true after *Create_Rules_K()* method is completed. Informatively, it

means that every time *Create_Rules_K()* method is called in the *while* loop, it builds the set of all confident rules, increasing the length of the rules by one. If the constructed set is not empty, Boolean variable *exist_new_rules* equals *true*. If building a new set of rules is impossible, the Boolean variable gets the value *false*. In this situation the cycle ends, thus completing the process of creating rules.

The predicate *Rules(k)* is described as follows: *the set of all frequent confident rules of length k + 1 is constructed at step k*.

This predicate is an invariant of the *while* loop. The truth of the predicate before the execution of the *while* loop guarantees the postcondition *Post1* of *Create_Rules_1()* method. *Create_Rules_K()* method preserves the truth of the predicate. At the end of the cycle it is guaranteed that the set of all frequent and confident rules is constructed, and it is impossible to extend the set. It is easy to show that the *while* loop will end since the length of rules is limited and cannot exceed the length of the longest record in the database.

For lemma 3 to be true in a particular implementation of the algorithm, it is necessary to build correct *Create_Rules_1()* and *Create_Rules_K()* methods in relation to the pre and postconditions. We will not give a specific implementation as it requires giving details of data presentation, including the necessary internal data. We only note that both methods are not complex, and programmers can easily create a specific implementation ensuring the truth of statements (18) and (19).

Create_Rules_1 method first calls another method to build the set of single frequent sets. The set is actively used in further constructions. In the double cycle a set of all frequent confident rules of length 1 is easily constructed by this set, which guarantees the truth of the postcondition of the method. Certainly, prior to calling *Create_Rules_1* it is necessary to call a method presenting the database in the desired format, thus ensuring fulfillment of the method's precondition.

Implementation of *Create_Rules_K* method is somewhat more complicated than that of the *Create_Rules_1* method. In its first call in the body of the *while* loop the precondition of the method is performed. It is guaranteed by the postcondition of *Create_Rules_1* method preceding the *while* loop in *Create_Rules* method. So that at calling *Create_Rules_K* method there exists a *current* set, i.e., the set of all frequent confident rules constructed at the previous step $k - 1$. In the cycle there is constructed an extension to each to the rules according to the set building rules, as described above. All confident rules with a single conclusion and a premise of length k not belonging to the premises of the previous step confident rules are added to the newly obtained confident rules. *Create_Rules_K* method preserves the invariant of the *while* loop and builds a new *current* set consisting of all frequent and confident rules of length $k + 1$. In case it is impossible to build the rules of this length, the process ends in cre-

ating a complete set of all frequent confident rules. It is guaranteed that the process will be completed.

Algorithm complexity

Let us evaluate the complexity of the basic methods used in *ConApriori* algorithm.

Complexity of Support method. The *Support* method calculating the frequency of the set, largely determines the efficiency of the whole algorithm, since all characteristics of the rules are defined through the *Support* function. For this reason, special attention should be given to both optimization of calculations of the method and the number of method calls. Presentation of sets and database records by *enumeration* given by *scale* allows us to reduce a complex operation of checking the occurrence of one set in another to a finite number of machine operations of the computer. Complexity of the operation is $O(1)$, whereas complexity of the *Support* method is $O(N)$, where N is the number of records in the database.

In implementing the algorithm we tried to minimize the number of possible calls of the function. Further, we consider two possible optimizations to reduce the time of computing the function significantly.

Complexity of Create_OBC_1 method. The method computes a set of single frequent items – *often_set_one*. The method is called only once in *Create_Rule_1* method constructing frequent confident rules of length 1.

The method is simple. For each property of *Properties* enumeration we compute its support (*Support*) and if it exceeds the minimum value, the property is included in the *often_set_one* set. It is clear that complexity of the method is $O(M * N)$, where M is the number of properties in the *Properties* enumeration.

Complexity of Create_OBC_K method. The method computes a set of the frequent sets of length k – *often_set_current*. The method is called only once in the *Create_Rule_K* method constructing frequent confident rules of length $k+1$.

The method builds pairs (*set*, *cand*), where *set* is an element of the *often_set_current* set built at the previous step, *cand* is an element of *often_set_one*. If the pair passes the *Apriori* test and can generate a frequent set, the *Support* function is calculated for it. Frequent pairs are included in set *often_set_current* of the next step.

Complexity of this method is $O(q * M * N)$, where q is the number of pairs having passed the *Apriori* test.

How large is this set? Consider a database of size N consisting of identical records, each of them containing all the elements of the *Properties* set of size M . Then any set will be frequent, any rule $X \rightarrow Y$ will be frequent and confident. The example is only of theoretical interest, since all the rules in this case are uninformative and therefore useless for practice. According to our estimates, when the record has the length an order less than M , the value of q is two to three orders less than N .

Complexity of Create_Rule_1 method. The method is called only once and builds the *current* set – a set of frequent confident rules of length 1. The method builds a rule with a single premise and a single conclusion. If the rule meets the required frequency and confidence, it is included in the *current* set.

Complexity of the method is $O(M2 * N)$, where M is the number of elements of the *often_set_one* set.

Complexity of Create_Rule_K method. The method is called repeatedly in the *while* loop of the main *Create_Rules* method building all the confident rules. At step k , it builds the *current* set of frequent confident rules of length $k + 1$ using the *current* set constructed at the previous step.

Complexity of the method is $O(NCR * M * N)$, where NCR is the number of elements of the *current* set.

Complexity of Create_Rules method. The method builds all frequent and confident rules and determines the final complexity of *ConApriori* algorithm. To evaluate its complexity is not difficult, as here in the *while* loop there is called *Create_Rule_K* method, whose complexity has already been evaluated. Let *iteration* be the number of iterations of the *while* loop. The final evaluation of the complexity of *Create_Rules* method and therefore of the entire *ConApriori* algorithm is:

$$O(iteration * NCR * M * N). \quad (21)$$

Among the four factors, the greatest is parameter N usually exceeding the product of all other factors, since the databases searched for Associative rules can be large.

Algorithm optimization

Optimization of the algorithm can be performed at different levels. To build an effective implementation, we can use low-level languages, such as Assembler or C, where we can write a code taking full advantage of the computer, which the code is built for. This is the way the effective Borglet's implementation [8] is constructed. Implementation in high-level languages, such as, for example, our implementation in the C# language, does not allow generating a code as efficient as a C language code, however it has such advantages as reusability, ease of understanding and modification of the code. Let us consider two possible optimizations applicable to our implementation of the algorithm.

Compressing the database. Usually database records are repeated. Therefore, it is possible to compress the database saving only the unique record and the number of its repetitions. If the average number of repetitions of records is equal to q , parameter N can be reduced to q times, thereby reducing the computational time by the same factor. This operation should be performed only when the average number of repetitions is greater than two.

Algorithm parallelization. Now that all computers are multi-core, we should parallelize the computa-

tion to increase the algorithm efficiency. In our case, it is advisable to parallelize the calculation of *Support* method. A database of N records can be divided into k groups and work with them in parallel. The number of groups can be adjusted to the number of cores on the computer the calculation is performed on. Parallelization of the *Support* method is transparent enough and does not cause any difficulties in the use of various tools of parallelism typical of modern programming languages.

This optimization has helped to speed up the computation about 4 times on a four-core computer. Using technologies, such as CUDA, and the GPU with its large number of cores we can achieve a much greater effect of parallelization.

Restrictions on the algorithm

Restriction on the data type. The algorithm assumes all data (object properties) to be binary – an object can either have a property or not. How can we overcome this limitation?

First, consider a property whose values belong to some interval. Such a property can be replaced with k binary properties. For example, a parameter characterizing a patient's temperature can be replaced with binary parameters: *normal*, *high*, *low*, *extra high*, *extra low*. The unnecessary details complicate detection of connections existing between items.

Consider a property of the discrete type that can take a finite number of values. Let these values be v_1, v_2, \dots, v_k . Then the property can be replaced with k binary properties. Whenever the source property takes the value v_i , the i -th binary property takes the value of 1, and other binary properties are set to 0.

Restriction on the number of properties. If we use the *enum* type defined in a programming language, the number of properties is limited for enumeration specified by scale. This limitation is connected with the limited interval of integer type the elements of the enumeration are mapped onto, for example, in C# the number of elements of the enumeration cannot be more than 64.

This limitation can be easily overcome. To implement the "long enumeration" we can write our own class *LongEnumScale*. That is exactly what we did in the project implementing the algorithm.

Note that, in our opinion, for meaningful associative rules, the number of properties should not be too large. For such classical applications of associative rules as market basket analysis you should look for multi-level associative rules, where the number of properties is limited at each level.

Details of implementation

The algorithm was implemented in C# in Visual Studio 2015 development environment. The Solution

includes four projects, namely DLL and three interface projects. The class library includes 6 classes.

- LongEnumScale – enumeration defined by a string of arbitrary length. The string is cut into k fragments, each being mapped onto the *ulong* integer type. K numbers will internally represent each transaction. The technique eliminates restrictions on the length of enumeration.

- Rule – a class defining properties and behavior of the objects presenting rules.

- ConAprioriLongRules – a class constructing frequent sets and confident rules.

- Apriori_DB – a class to work with database. The *Support* method is included in the class.

- Model – a class allowing creation of a model database.

- Converter – a class whose methods allow us to convert the original database into the format required for the algorithm.

Three projects define a user-friendly interface, which allows: creating the model database, converting the database, building confident rules, filtering and saving the rules, as well as building a specific rule.

Approbation and efficiency of the algorithm

Comparison of ConApriori with Microsoft Analysis Services and Arules tools. *ConApriori* algorithm was implemented to solve a real problem of medical diagnosis [13]. For physicians the discovered rules had a scientific interest that allowed them to confirm the hypothesis of a possible early diagnosis of metabolic syndrome. The rules had a high degree of confidence and correlation.

The article [14] shows a comparison of the effectiveness of *ConApriori* Toolkit with two widely used tools for building Associative rules.

- *Microsoft Analysis Services* Tools [15, 16] allows constructing Associative rules for data prepared in Excel and then processed in SQL Server.

- *Arules* Tools are built into RStudio, a popular development environment for the R language [17]. The *Arules* package includes one of the earliest versions of *Apriori* algorithm developed by Christian Borgelt.

In contrast to *ConApriori* algorithm both tools – *Microsoft Analysis Services* and *arules* – can only build rules with a single conclusion.

In most cases the time of constructing rules was minimal for Borgelt's version. However, reading a database implemented by means of the R language, substantially slowed down the whole process of creating rules. A rather complex process of building the rules used in *Microsoft* tools demanded more time in comparison with other algorithms and occasionally led to incorrect results.

It should be noted that the current version of *ConApriori* algorithm has been significantly improved in comparison with the version considered in [14].

Comparison of current ConApriori version with current Borgelt's version. Let us compare the efficiency of the *ConApriori* algorithm and the *Apriori* algorithm developed by Christian Borgelt. Borgelt's algorithm used version 6.24 as of 15.10. 2016 available on his website [8], written in C and having been developed for 20 years. The version is an *exe* file being run from the command line.

All the following results are obtained on a computer with an Intel Core I7, 2.20 GHz processor.

Model database. Using our toolkit we created two model databases of 100,000 and 500,000 records. The number of items in the records is 100. They are divided into four groups of 50, 25, 15 and 10 items. The probability of occurrence of the elements in each group is equal to 0.05, 0.1, 0.15 and 0.7, respectively. The maximum length of transactions is 10.

High probability of occurrence of the forth group transaction items implies appearance of the corresponding associative rules. Table 1 shows the results of calculations for *ConApriori* algorithm for a database of 100,000 transactions.

The last two columns of the table show the number of rules and the time of their construction.

Table 1

Model database 100,000. ConApriori algorithm

Confidence_min	Support_min	N_Rules	T_rules (sec.)
0,3	0,1	90	0,38
0,3	0,05	447	1,7
0,3	0,03	450	1,7
0,3	0,01	1 290	6
0,15	0,03	704	2,3

The time of reading the database of 100,000 transactions is 0.12 seconds.

Table 2 gives the results for Borglet's algorithm. The last column shows the number of nodes in the transaction tree.

Table 2

Model database 100,000. Borglet's Apriori algorithm

Confidence_min	Support_min	N_Rules	T_rules (sec.)	N nodes in Transaction tree
0,3	0,1	90	< 0,01	1 024
0,3	0,05	447	< 0,01	1 024
0,3	0,03	450	< 0,01	1 024
0,3	0,01	1290	0,03	44 717
0,15	0,03	450	< 0,01	1 024

The time of reading the database of 100,000 transactions is 0.05 seconds.

Tables 3 and 4 show similar results for the database containing 500,000 transactions.

Table 3

Model database 500,000. ConApriori algorithm

Confidence_min	Support_min	N_Rules	T_rules (sec.)
0,3	0,1	90	1,7
0,3	0,05	450	8,2
0,3	0,03	450	8,2
0,3	0,01	1290	26,4
0,15	0,03	704	10,4

The time of reading the database of 500,000 transactions is 0.6 seconds.

Table 4

**Model database 500,000.
Borglet's Apriori algorithm**

Confidence_min	Support_min	N_Rules	T_rules (sec.)	N nodes in Transaction tree
0,3	0,1	90	< 0,01	1 024
0,3	0,05	447	< 0,01	1 024
0,3	0,03	450	< 0,01	1 024
0,3	0,01	1 290	0,11	157 741
0,15	0,03	450	< 0,01	1 024

The time of reading the database of 500,000 transactions is 0.24 seconds.

Let us analyze the obtained results. *ConApriori* algorithm demonstrates the efficiency acceptable for practical use. For a database of 100,000 transactions it detects 90 rules in less than one second. The time dependence of the algorithm on the database size and the number of constructed rules is linear.

Borglet's algorithm shows hyper efficiency. It builds and processes 157,000 nodes of the transactions tree in tenths of a second. This high efficiency is due to two factors, namely the quality of the algorithm and its implementation in the C language.

As is noted above, the algorithm does not build rules with multiple conclusions. The number of rules can be very different. For example, *ConApriori* algorithm builds 704 rules, whereas the number of the rules built with Borglet's algorithm is 450, being limited to the rules with a single conclusion. For the latter rules both algorithms build the same set of rules.

We can also note that a database of 100,000 transactions is as representative as the base of 500,000 records. In fact, both databases use the same set of rules, with only one case showing the minimal variance of three rules.

FIMI website databases. The FIMI website [18] is a database repository (Frequent Itemset Mining Dataset Repository) used for comparative analysis of efficiency of algorithms building frequent examples. The databases presented on the website contain records in *csv* format. The meaning of the elements in the records is not specified, each element is represented by a numeric code. The number of records in the database and the number of different elements can be quite large. For example, in *Kosarak* database the number of records is more than 990,000, the number of different elements being more than 10,000.

For algorithms such as *ConApriori*, whose task is to build confident Associative rules, the direct use of databases like *Kosarak* is hardly reasonable, since there is no point in looking for associations, the number of different elements being so great. Elements should be combined into categories, building multi-level Associative rules. This approach was applied to the *Kosarak* database analysis regarding the first two digits

of the element code as the category code. In this case associative rules specify the connection between 99 categories of elements.

It took *ConApriori* algorithm about 0.6 seconds to find 13 rules with the minimum support of 0.1 and minimum confidence of 0.5.

In the same environment *Apriori* algorithm builds a tree of transactions from 512 nodes, in the time commensurate with the measurement error. It builds one rule fewer without detecting the rule with multiple conclusions.

Consider the results for another database from the website – the *accidents* database. The database built, as is noted, on real data, contains 340,183 transactions. The number of different elements is 468.

It took *ConApriori* algorithm 0.6 seconds to find 329 rules with the minimum support of 0.85 and minimum confidence of 0.9.

In the same environment *Apriori* algorithm builds a tree of transactions from 140 nodes, as usual, in the minimum time commensurate with the measurement error. Nevertheless, it builds 156 rules – much fewer than *ConApriori* does — as more than half of the 329 rules are those with multiple conclusions.

It should be noted that despite the high frequency and confidence of the constructed rules, they are informatively useless as the *lift* parameter for all discovered rules is practically equal to one, which indicates the absence of any correlation between the premise and conclusion of the rules. This has been expected, as the number of items contained in the transactions is great and detectable associations are random. In such situations it is wiser to search multi-level rules by combining the elements in groups.

Wiki datasets databases. Databases allowing us to test algorithms for constructing frequent examples and Associative rules are also available on the Wiki website [19]. Let us do the research on the databases containing real data about purchases of confectionery. The site contains four databases of this kind. The number of records in the databases are 1 000, 5 000, 20 000 and 75 000. The number of different elements found in the records is 50, among them 40 confectionery products and 10 drinks of different kinds. Elements in the records are represented by codes of the products, but a meaningful decryption of the codes is supplied including the name, price and other product characteristics.

The databases present a classic pattern of market baskets to search for Associative rules. They ideally suit the effective search for rules by our algorithms. All 50 elements found in the records belong to the same category, so the databases provide a good example of search for associations within one product category. The fact that not only one database but four bases with the same elements are presented, allows us not only to detect the rules, but also to test the hypothesis of stability of the rules, i.e., to see the degree of coincidence of the rules built on all four databases representing the same population.

Table 5 presents the results of applying *ConApriori* algorithm to the four databases.

Table 5

ConApriori algorithm

Confidence_min	Support_min	N_Rules	T_rules (sec.)	N transactions
0,45	0,03	75	0,05	1 000
0,45	0,03	65	0,08	5 000
0,45	0,03	45	0,2	20 000
0,45	0,03	42	0,6	75 000

Table 6 presents the results of applying *Apriori* algorithm to the same databases.

Table 6

Apriori algorithm

Confidence_min	Support_min	N_Rules	T_rules (sec.)	N transactions	N nodes
0,45	0,03	64	<0,01	1 000	874
0,45	0,03	55	<0,01	5 000	3 783
0,45	0,03	41	<0,01	20 000	12 551
0,45	0,03	41	0,01	75 000	38 842

The effectiveness of the search for the rules in the databases is high for both algorithms. Since Borglet's algorithm does not build rules with multiple conclusions, the number of detectable rules is somewhat less than for *ConApriori* algorithm. However, all the rules built coincide with those built by *ConApriori* algorithm.

What to say about the stability of the discovered rules? Out of the 42 rules discovered in the database of 75,000 records, 30 rules meet the 4 of 4 criteria, that is, they are present in the rules constructed for all four databases. The remaining 12 rules meet the 3 of 4 criteria. These rules are not found in the database of 1000 records. The database has more random associations.

The following table provides examples of stable rules occurring in all databases.

Table 7

Examples of stable rules

The number of records in database	Rules
1 000	33 => 42 support = 0,04 confidence = 0,49 lift = 5,94 3, 18 => 35 support = 0,04 confidence = 0,93 lift = 12,36
5 000	33 => 42 support = 0,04 confidence = 0,56 lift = 6,06 3, 18 => 35 support = 0,04 confidence = 0,94 lift = 10,49
20 000	33 => 42 support = 0,04 confidence = 0,54 lift = 5,82 3, 18 => 35 support = 0,04 confidence = 0,94 lift = 10,13
75 000	33 => 42 support = 0,04 confidence = 0,52 lift = 5,72 3, 18 => 35 support = 0,04 confidence = 0,95 lift = 10,24

We can see that stability of rules and coincidence of characteristics confirms the hypothesis that the constructed rules are a reflection of the real existing relations between elements.

Table 8 shows decrypting the codes of the items included in the rules presented in table 7.

Table 8

Characteristics of the products

The code of element	The code's interpretation
33	'Cheese','Croissant',1.75,'Food'
42	'Orange','Juice',2.75,'Drink'
3	'Opera','Cake',15.95,'Food'
18	'Cherry','Tart',3.25,'Food'
35	'Apricot','Danish',1.15,'Food'

One of the rules says that choosing a cheese croissant is often accompanied by choosing orange juice. Such information can be useful for providing the best possible service to the customers of the bakery.

Do we need the rules with multiple conclusions?

In contrast to the known tools (Borglet's algorithm, Microsoft Toolkit), *ConApriori* builds rules with a multiple conclusion.

Explaining why his algorithm does not build rules with multiple conclusions, Christian Borglet writes: "Multiple items in the consequents of association rules come at a considerable cost, which must be counter-balanced by some added benefit in order to justify them".

He rightly notes that the existence of the confident rule with a multiple conclusion, $x \rightarrow y, z$ implies the existence of four confident rules with a single conclusion: $x \rightarrow y$; $x \rightarrow z$; $x, y \rightarrow z$; $x, z \rightarrow y$. Further he writes: "Given that a customer placed the item x into its shopping cart, we first use the rule $x \Rightarrow y$ to infer that the customer is likely interested in y , then hypothesize that the customer will place item y into the shopping, and apply the rule $x, y \rightarrow z$ to the resulting situation to suggest z as well. In this way we achieve the same result, without any loss of information or approximation, that would result from $x \rightarrow y, z$ ".

The latter statement is hard to accept. The existence of confident rules with a single conclusion does not imply the existence of a confident rule with multiple conclusions. Meanwhile, in practice, such rules may be of particular interest. When a Bank or some other organization offers the customers a set of services, it is important to understand not only which customers prefer a particular service, but also which customers choose a package of services.

Currently together with a group of physicians we are conducting a research of the risk factors affecting the emergence of children's diseases. It is important to discover the rules giving evidence of the factors influencing the possibility of occurrence of a certain set of diseases.

Similar situations (customers and the services offered, patients and possible diseases) can be considered as the tasks of classification. For them the reduction in the number of constructed rules is due to the fact that the premise of the rules is chosen from one set, and the conclusion is taken from another one.

Conclusion

ConApriori algorithm is based on presenting the database transaction using enumeration given by scale. Database records in this case can be presented by one or more integers. The basic algorithm for the *Support* function is computed in linear time proportional to the size of the database, with the minimum constant. Using capabilities of parallelization we can sufficiently reduce the computation time of this function.

The algorithm provides high performance, practically sufficient for building associative rules in large databases.

It would be of interest to study the possibility of combining the approach used in the algorithm with methods of compressing the database into a tree of transactions.

Acknowledgements. I am grateful to Michael Dekhtyar and Alexander Dekhtyar for their careful reading of the manuscript and important comments for improving the text of the article. My thanks to my sons, Ilya and Yuly Billig for their valuable comments. Many thanks to Nicholas Siabro – we worked together on the experiments on comparative performance analysis of various algorithms. And finally, I would like to specially thank Christian Borglet for the possibility of having free access to the code of his excellent algorithm which he has been developing for many years.

References

1. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases. *Proc. of the ACM SIGMOD Conf. on Management of Data*. 1993.
2. Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases. *Proc. 20th Int. Conf. on Very Large Data Bases (VLDB)*. 1994.
3. Agrawal R., Srikant R. Mining sequential patterns. *Proc. 11th Int. Conf. Data Engineering*. IEEE Press, 1995.
4. Bodon F. Fast Apriori Implementation. *Proc. IEEE ICDM Workshop on Frequent Itemset Mining Implementations*. 2003.
5. Borgelt Ch., Kruse R. Induction of Association Rules: Apriori Implementation. *Proc. 15th Conf. on Computational Statistics (Compstat 2002)*. Physica Verlag Publ., Heidelberg, Germany, 2002, pp. 395–400.
6. Borgelt Ch. Efficient Implementations of Apriori and Eclat. *Workshop of Frequent Item Set Mining Implementations (FIMI 2003)*. Melbourne, FL, USA, 2003.
7. Borgelt Ch. Recursion Pruning for the Apriori Algorithm. *2nd Workshop of Frequent Item Set Mining Implementations (FIMI 2004)*. Brighton, UK, 2004.
8. Ch. Borglet's web pages. Available at: <http://www.borgelt.net/apriori.html> (accessed March 23, 2017).
9. Han J., Fu Y. Mining Multiple-Level Association Rules in Large Database. *IEEE transactions on knowledge & data engineering in 1999*, pp. 1–12.
10. Shrivastava A., Jain R.C. Performance Analysis of Modified Algorithm for Finding Multilevel Association Rules. *Computer Science & Engineering: An Int'l Jour. (CSEIJ)*. 2013, vol. 3, no. 4.
11. Borgelt Ch. Frequent Item Set Mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2012, no. 2(6), pp. 437–456, J. Wiley & Sons Publ., Chichester, UK.
12. Han J., Kamber M., Pei J. *Data Mining. Concepts and Techniques*. Elsevier Publ., 2012.
13. Billig V.A., Ivanova O.V., Tsaregorodtsev N.A. Building association rules in medical diagnosis. *Programmnye produkty i sistemy [Software & Systems]*. 2016, no. 2. Available at: <http://www.swsys.ru/index.php?page=article&id=4162> (accessed March 23, 2017).
14. Billig V.A., Korneeva E.I., Syabro N.A. Association rules. Compared analysis of the tools. *Programmnye produkty, sistemy i algoritmy [Software Products, Systems and Algorithms]*. 2016, no. 2. Available at: <http://swsys-web.ru/association-rules.html> (accessed March 23, 2017).
15. MacLennan J., Tang Z., Crivat B. *Data Mining with Microsoft SQL Server 2008*. Wiley Publ., 2009, 676 p.
16. *Search associations Microsoft*. Available at: [https://msdn.microsoft.com/ru-ru/library/ms174916\(v=sql.105\).aspx](https://msdn.microsoft.com/ru-ru/library/ms174916(v=sql.105).aspx) (accessed March 23, 2017).
17. Ihaka R., Gentleman R. R: A Language for Data Analysis and Graphics. *Jour. of Computational and Graphical Statistics*. 1996, vol. 5, no. 3.
18. *Examples of the databases*. Available at: <http://fimi.ua.ac.be/data/> (accessed March 23, 2017).
19. *Examples of the databases*. Available at: <https://wiki.csc.calpoly.edu/datasets/wiki/WikiStart> (accessed March 23, 2017).

УДК 004.89

DOI: 10.15827/0236-235X.118.196-206

Дата подачи статьи: 16.03.17

2017. Т. 30. № 2. С. 196–206

ЭФФЕКТИВНЫЙ АЛГОРИТМ КОНСТРУИРОВАНИЯ АССОЦИАТИВНЫХ ПРАВИЛ

В.А. Биллиг, к.т.н., старший научный сотрудник, доцент, Vladimir-Billig@yandex.ru
(Тверской государственный технический университет,
наб. Аф. Никитина, 22, г. Тверь, 170026, Россия)

Конструирование ассоциативных правил является одним из наиболее важных алгоритмов извлечения знаний из БД. Все современные алгоритмы так или иначе связаны с алгоритмом Apriori, предложенным в работах Р. Агравала и его соавторов, опубликованных более чем 20 лет назад и ставших сегодня классикой. Известные эффективные реализации алгоритма связаны со сжатием БД и представлением структуры данных в виде дерева, что позволяет эффективно вычислять поддержку ассоциативных правил и другие их характеристики.

Предлагаемый алгоритм ConApriori не использует вышеназванную идею. Транзакции БД рассматриваются как пересечение, заданное шкалой. Это позволяет практически мгновенно вычислять базисную для алгоритма операцию, определяющую, является ли некоторое множество подмножеством другого множества. Вычисления сводятся к не-

скольким логическим командам компьютера. Перечисление позволяет также рассматривать транзакцию во внутреннем представлении как одно или несколько чисел, сохраняя в то же время смысл элементов транзакции в их внешнем представлении.

Другая идея, используемая в алгоритме, позволяет конструировать большинство достоверных правил на основе ранее построенных достоверных правил.

В статье дается обоснование корректности алгоритма и приводится оценка его сложности. Анализируется эффективность алгоритма в сравнении с другими известными реализациями. Рассматривается также возможность распараллеливания алгоритма.

Ключевые слова: *Data Mining, априори, ассоциативные правила, поддержка, достоверность, лифт, перечисление, шкала, БД, транзакция, обнаружение знаний, параллельные вычисления, корректность, сложность, антимонотонность.*

Литература

1. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases. Proc. of the ACM SIGMOD Conf. on Management of Data. 1993.
2. Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases. Proc. 20th Int. Conf. on Very Large Data Bases (VLDB). 1994.
3. Agrawal R., Srikant R. Mining sequential patterns. Proc. 11th Int. Conf. Data Engineering. IEEE Press, 1995.
4. Bodon F. Fast Apriori Implementation. Proc. IEEE ICDM Workshop on Frequent Itemset Mining Implementations. 2003.
5. Borgelt Ch., Kruse R. Induction of Association Rules: Apriori Implementation. Proc. 15th Conf. on Computational Statistics (Compstat 2002). Physica Verlag Publ., Heidelberg, Germany, 2002, pp. 395–400.
6. Borgelt Ch. Efficient Implementations of Apriori and Eclat. Workshop of Frequent Item Set Mining Implementations (FIMI 2003). Melbourne, FL, USA, 2003.
7. Borgelt Ch. Recursion Pruning for the Apriori Algorithm. 2nd Workshop of Frequent Item Set Mining Implementations (FIMI 2004). Brighton, UK, 2004.
8. Ch. Borgelt's web pages. URL: <http://www.borgelt.net/apriori.html> (дата обращения: 23.03.2017).
9. Han J., Fu Y. Mining Multiple-Level Association Rules in Large Database. IEEE transactions on knowledge & data engineering in 1999, pp. 1–12.
10. Shrivastava A., Jain R.C. Performance Analysis of Modified Algorithm for Finding Multilevel Association Rules. Computer Science & Engineering: An Int. Jour. (CSEIJ). 2013, vol. 3, no. 4.
11. Borgelt Ch. Frequent Item Set Mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2012, no. 2 (6), pp. 437–456, J. Wiley & Sons Publ., Chichester, UK.
12. Han J., Kamber M., Pei J. Data Mining. Concepts and Techniques. Elsevier Publ., 2012.
13. Биллиг В.А., Иванова О.В., Царегородцев Н.А. Построение ассоциативных правил в задаче медицинской диагностики // Программные продукты и системы. 2016. № 2. URL: <http://www.swsys.ru/index.php?page=article&id=4162> (дата обращения: 23.03.2017).
14. Биллиг В.А., Корнеева Е.И., Сябро Н.А. Ассоциативные правила. Сравнительный анализ инструментария // Программные продукты, системы и алгоритмы. 2016. № 2. URL: <http://swsys-web.ru/association-rules.html> (дата обращения: 23.03.2017).
15. MacLennan J., Tang Z., Crivat B. Data Mining with Microsoft SQL Server 2008. Wiley Publ., 2009, 676 p.
16. Search associations Microsoft. URL: [https://msdn.microsoft.com/ru-ru/library/ms174916\(v=sql.105\).aspx](https://msdn.microsoft.com/ru-ru/library/ms174916(v=sql.105).aspx) (дата обращения: 23.03.2017).
17. Ihaka R., Gentleman R. R: A Language for Data Analysis and Graphics. Jour. of Computational and Graphical Statistics. 1996, vol. 5, no. 3.
18. Examples of the databases. URL: <http://fimi.ua.ac.be/data/> (дата обращения: 23.03.2017).
19. Examples of the databases. URL: <https://wiki.csc.calpoly.edu/datasets/wiki/WikiStart> (дата обращения: 23.03.2017).

Примеры библиографического описания статьи

1. Billig V.A. Effective algorithm for constructing associative rules // Программные продукты и системы. 2017. Т. 30. № 2. С. 196–206; DOI: 10.15827/0236-235X.118.196-206.
2. Billig V.A. Effective algorithm for constructing associative rules. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 196–206 (in Russ.); DOI: 10.15827/0236-235X.118.196-206.

УДК 004.93
DOI: 10.15827/0236-235X.118.2.207-216

Дата подачи статьи: 03.04.17
2017. Т. 30. № 2. С. 207–216

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ ПРИ РЕАЛИЗАЦИИ WEB-ИНСТРУМЕНТАРИЯ РАСПОЗНАВАНИЯ ОБРАЗОВ НА ОСНОВЕ МЕТОДОВ ПРЕЦЕДЕНТОВ

*В.В. Фомин, д.т.н., профессор, v_v_fomin@mail.ru;
И.В. Александров, аспирант, chrono555@yandex.ru
(Российский государственный педагогический университет им. А.И. Герцена,
наб. реки Мойки, 48, г. Санкт-Петербург, 191186, Россия)*

Предлагается программное решение, направленное на улучшение качества распознавания образов и повышение эффективности инструментария машинного обучения посредством реализации grid-технологий. Формулируются стратегические направления разработки инструментария распознавания образов в виде программной системы, основанной на принципах распределенных систем, распараллеливания и адаптивной настройки вычислительных ресурсов. Рассматривается структура организации web-инструментария распознавания образов с использованием концепции библиотеки алгоритмов. Даются алгоритмические решения распараллеливания алгоритмов обучения и распознавания на базе классических методов интеллектуального анализа данных, которые хорошо зарекомендовали себя на практике. К таким алгоритмам относятся методы прецедентов или методы, основанные на метриках близости. Они обладают большим потенциалом к распараллеливанию вычислительных процессов и разработке параллельных алгоритмов их реализации.

Поиски путей повышения производительности вычислительной техники, особенно при реализации web-инструментария на основе ресурсоемких вычислительных алгоритмов машинного распознавания и прогнозирования, привели к решению создать grid-систему. Рассматриваемые в статье архитектура и реализация grid-системы предполагают распараллеливание и организацию распределенных вычислений на многомашинной основе с применением интернет-технологий, что позволяет получить практически те же вычислительные мощности, что и на многопроцессорных компьютерных системах, но с гораздо меньшей стоимостью.

В статье решается задача повышения эффективности вычислительных ресурсов при возможности реконфигурирования структуры интернет-соединений, в том числе процедуры настройки структуры вычислительной сети, подключаемых каналов связи и выделенных серверов в зависимости от исходных алгоритмов и данных. Представлены зависимости параметров времени выполнения операций от дисциплины обслуживания, адаптирующей систему к запросам пользователей. При этом задачи ранжируются по ресурсоемкости и под них выделяются соответствующие их рангу вычислительные мощности.

Ключевые слова: интеллектуальные информационные системы, распознавание образов, машинное обучение, web-системы, параллельные вычисления.

В современном мире рост производительности при обработке большого количества разнообразной информации достигается только в тех случаях, когда часть интеллектуальной нагрузки берут на себя компьютеры. Одним из направлений исследований в области искусственного интеллекта [1, 2] являются интеллектуальные информационные системы, которые находятся на пике исследовательской активности, практики прикладного применения и коммерциализации. Они широко вошли в разные области промышленности, экономики, медицины и т.д. Интеллектуальные информационные системы сосредоточили в себе наиболее наукоемкие технологии с высоким уровнем автоматизации процессов не только подготовки информации для принятия решений, но и выработки вариантов решений, опирающихся на полученные информационной системой данные.

Проблемы эффективного применения, использования, внедрения интеллектуальных систем определяют сложность подходов к классификации задач искусственного интеллекта и решаются в рамках различных направлений исследований, основанных на близких моделях, методах и алгоритмах. К таким исследованиям относятся распознава-

ние образов, машинное обучение, машинный перевод, интеллектуальный анализ данных и др. [2].

Одной из базовых апорий интеллектуального анализа данных [3] являются технологии распознавания образов [4–6]. Распознавание образов – класс задач, связанных с определением принадлежности объекта к одному из классов объектов. Распознавание образов можно рассматривать как классификацию объектов. Сложность и многофакторность проблемы распознавания образов, разнообразие проблемных областей и категорий задач, отсутствие универсальных методов решения задач распознавания приводят к тому, что традиционная методология разработки систем распознавания, как правило, не позволяет достигнуть требуемой эффективности результатов в рамках промышленного применения.

В настоящее время существует значительное противоречие, обусловленное уровнем развития формального аппарата теории распознавания образов, сравнительной ограниченностью и узкой направленностью большинства из существующих прикладных систем распознавания. Данное противоречие частично может быть преодолено путем разработки интегрированных многофункциональ-

ных автоматизированных информационных систем распознавания, ориентированных на решение широкого класса задач и допускающих адаптацию и настройку на новую предметную область и категорию задач.

Описание программы. В рамках исследований по тематике «Web-инструментарий машинного обучения» в качестве прототипа облачного ресурса разработана программа «Web-инструментарий распознавания образов на основе расширяющейся библиотеки методов машинного обучения» [7, 8]. Программа может быть использована для принятия решений широким кругом лиц, в том числе студентами, преподавателями, бизнесменами, специалистами в различных областях.

В базовой версии программы реализован класс методов прецедентов как пользующаяся популярностью классика в области data-mining. Методы прецедентов состоят из следующих циклов: 1) извлечение похожего прецедента из библиотеки; 2) использование выбранного прецедента для построения решения; 3) проверка полученного решения; 4) занесение решения как нового прецедента в библиотеку. Таким образом, системой накапливается опыт (прецеденты) и реализуется машинное обучение, что способствует повышению качества принимаемого решения [9].

Главным достоинством методов прецедентов является их высокая эффективность для множества задач с точки зрения достоверности результата и компактности алгоритмов анализа, а главным недостатком – требование значительных вычислительных ресурсов. Особенно это актуально при работе с большими объемами данных.

В качестве основных алгоритмов анализа в программе (рис. 1) были реализованы такие методы прецедентов, как метод k-ближайших соседей, метод потенциальных функций, метод эталонов, метод коррекционных приращений, метод наименьших среднеквадратических ошибок, метод наивный байесовский. Методы основаны на алгоритмах

метрической классификации и оценивании сходства объектов, для всех методов характерно наличие обучающей выборки.

Программа развивает концепцию облачных вычислений и осуществляет распознавание образов на основе методов машинного обучения посредством реализации web-сервиса и использования web-технологий. Компоненты web-технологий в программе реализованы с помощью сервлетов, которые расширяют функциональность web-серверов. Выбор в пользу сервлетов обусловлен тем, что они позволяют создавать платформенно-независимый код, предоставляют возможность масштабирования программы и могут выполняться параллельно в рамках одного процесса на сервере.

Алгоритмы распараллеливания методов прецедентов. Борьба за эффективное использование вычислительных ресурсов привела к появлению технологий распределенных систем и параллельных вычислений [10, 11], в частности, grid- и интернет-технологий. Grid-система строится по принципу клиент–сервер, состоит из одного или нескольких компьютеров-серверов и множества компьютеров-клиентов свободной конфигурации и представляет собой эффективную для распараллеливания структуру вычислительных ресурсов.

Так как у методов прецедентов наличие обучающей выборки обязательно, в их работе можно выделить два последовательных этапа: этап обучения и этап распознавания. Рассмотрим схемы реализации этих этапов на одном компьютере (базовом для расчетов компьютере K_0) более подробно. В рамках исследований множества обучающих и классифицируемых объектов представлены как отдельные файлы, а объекты в них – как строки атрибутов. Файл с обучающими объектами ($F_{\text{обуч.}}$) содержит w объектов (строк), а файл с классифицируемыми объектами ($F_{\text{классиф.}}$) имеет m объектов.

Этап обучения. Этап обучения на K_0 компьютере заключается в следующем: 1) компьютер K_0 поочередно считывает один из w обучающих объектов (q_k -й объект) из файла $F_{\text{обуч.}}$ и подает его на вход классификатора; 2) классификатор, получив q_k -й обучающий объект, строит на его основе решение с учетом меры близости применяемого метода прецедента. После построения решения на основе всех w объектов из файла $F_{\text{обуч.}}$ на выходе классификатора образуется система решающих правил, способная определить класс классифицируемого объекта.

Обозначим время обучения классификатора на q_k -м объекте как $T_{\text{обуч.}}(q_k)$. Тогда время обучения на основе файла $F_{\text{обуч.}}$ на K_0 компьютере рассчитывается как суммарное время

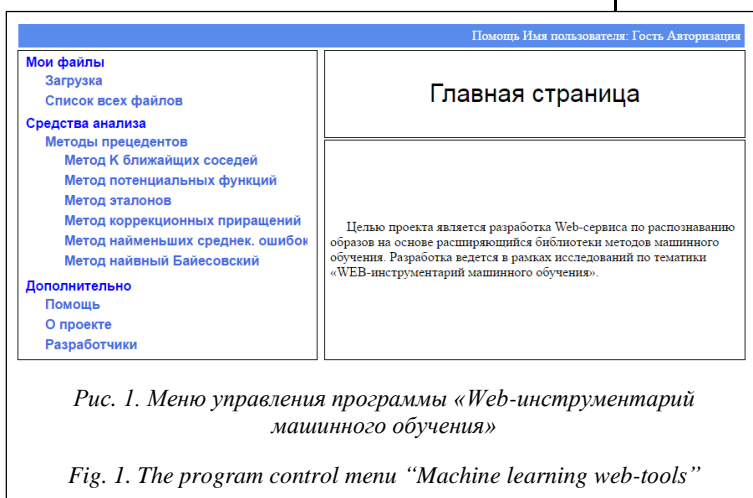
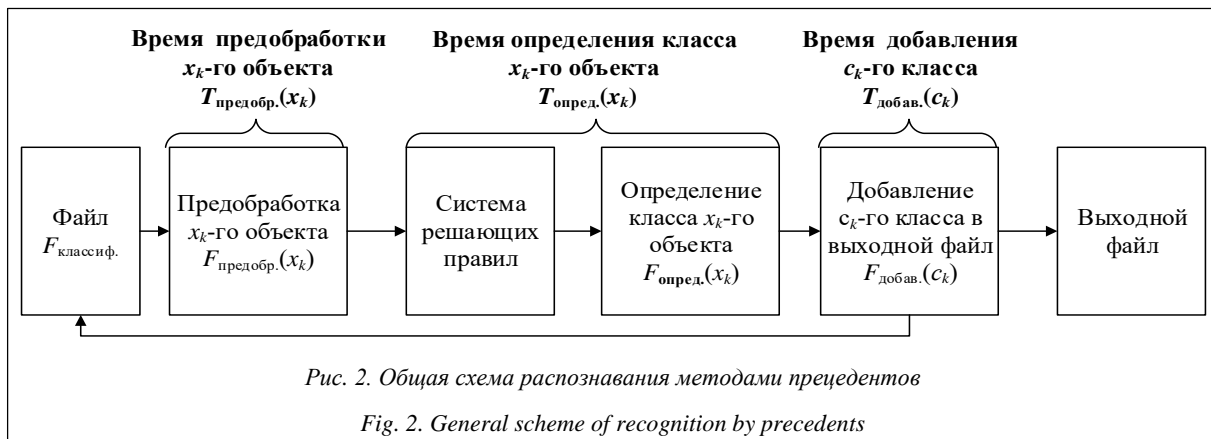


Рис. 1. Меню управления программы «Web-инструментарий машинного обучения»

Fig. 1. The program control menu “Machine learning web-tools”



обучения классификатора на w обучающих объектах:

$$T_{0 \text{ обуч.}} = \sum_{k=1}^w T_{0 \text{ обуч.}}(q_k). \quad (1)$$

Этап распознавания. Этап распознавания на K_0 компьютере (рис. 2) заключается в следующем: 1) компьютер K_0 поочередно занимается предобработкой одного из m классифицируемых объектов (x_k -й объект) из файла $F_{\text{классиф.}}$; обозначим этот этап как функцию $F_{\text{предобр.}}(x_k)$, а время, затраченное на предобработку x_k -го объекта, как $T_{\text{предобр.}}(x_k)$; 2) затем x_k -й объект поступает на вход системы решающих правил, которая определяет его класс (c_k); обозначим этот этап как функцию $F_{\text{опред.}}(x_k)$, а время, затраченное на определение класса x_k -го объекта, как $T_{\text{опред.}}(x_k)$; 3) в заключение класс x_k -го объекта (c_k) добавляется в выходной файл, этот этап обозначим как функцию $F_{\text{добав.}}(c_k)$, а время, затраченное на добавление c_k -го класса, как $T_{\text{добав.}}(c_k)$.

Тогда время распознавания на K_0 компьютере рассчитывается по формуле

$$T_{0 \text{ расп.}} = \sum_{k=1}^m (T_{0 \text{ предобр.}}(x_k) + T_{0 \text{ опред.}}(x_k) + T_{0 \text{ добав.}}(c_k)). \quad (2)$$

Или, сгруппировав время по функциям, перепишем формулу (2):

$$T_{0 \text{ расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \sum_{k=1}^m T_{0 \text{ опред.}}(x_k) + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (3)$$

Рассмотрим и проанализируем входящие в распознавание классифицируемых объектов процессы на K_0 компьютере более подробно.

Процесс предобработки (функция $F_{\text{предобр.}}(x_k)$) на K_0 компьютере состоит из двух этапов: 1) чтение x_k -го объекта из файла $F_{\text{классиф.}}$; обозначим чтение как функцию $F_{\text{чтен.}}(x_k)$, а время, затраченное на чтение x_k -го объекта, как $T_{\text{чтен.}}(x_k)$; 2) запись x_k -го объекта в выходной файл; обозначим запись как функцию $F_{\text{зап.}}(x_k)$, а время, затраченное на запись прочитанного x_k -го объекта в выходной файл, как $T_{\text{зап.}}(x_k)$. Другими словами, время предобработки x_k -го объекта на K_0 компьютере определяется по формуле $T_{0 \text{ предобр.}}(x_k) = T_{0 \text{ чтен.}}(x_k) + T_{0 \text{ зап.}}(x_k)$.

Процесс определения класса специфичен для каждого из методов прецедентов, следовательно, время определения класса будет зависеть от примененного метода прецедента.

Процесс добавления результатов распознавания x_k -го объекта в виде класса c_k в выходной файл (функция $F_{\text{добав.}}(c_k)$) на одном K_0 компьютере представляет собой функцию записи класса c_k в конец выходного файла и рассчитывается следующим образом: $T_{0 \text{ добав.}}(c_k) = T_{0 \text{ зап.}}(c_k)$.

Дополнительно введем два параметра: $T_{0 \text{ средн. опред.}}$ и $V_{0 \text{ средн. опред.}}$.

$T_{0 \text{ средн. опред.}}$ — среднее время определения класса на K_0 компьютере одного классифицируемого объ-

$$\text{екта: } T_{0 \text{ средн. опред.}} = \frac{\sum_{k=1}^m T_{0 \text{ опред.}}(x_k)}{m}.$$

$V_{0 \text{ средн. опред.}}$ — средняя скорость определения класса объектов на K_0 компьютере, то есть сколько классов объектов (число может быть нецелым) K_0 компьютер определяет в единицу времени:

$$V_{0 \text{ средн. опред.}} = \frac{1}{T_{0 \text{ средн. опред.}}}. \quad (4)$$

Стоит отметить, что $T_{0 \text{ средн. опред.}}$ и $V_{0 \text{ средн. опред.}}$ для всех объектов распознавания на K_0 компьютере одинаковы, потому что порядок и количество операций для определения класса любого объекта одинаково, а скорость выполнения операции на K_0 компьютере постоянна. Следовательно, чем больше скорость выполнения операции на K_0 компьютере (производительность), тем меньше $T_{0 \text{ средн. опред.}}$ и больше $V_{0 \text{ средн. опред.}}$.

С учетом формулы (4) формулу (3) можно переписать так:

$$T_{0 \text{ расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{m}{V_{0 \text{ средн. опред.}}} +$$

$$+ \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (5)$$

Реализация параллельной вычислительной системы. Анализ основных процессов этапа распознавания и формулы (2) позволяет сделать вывод, что распознавание x_k -го объекта никак не связано с распознаванием остальных классифицируемых

объектов. Таким образом, распознавание объектов можно делать параллельно на n компьютерах, а записывать результаты распознавания в один выходной файл на центральном сервере. Другими словами, подключив n компьютеров к K_0 , можно получить вычислительную сеть, где K_0 будет играть роль центрального сервера.

На рисунке 3 показана структура такой параллельной вычислительной системы из n компьютеров K_1, K_2, K_n и сервера K_0 , где функции $F_{\text{предобр.}}(x_k)$, $F_{\text{добав.}}(c_k)$ такие же, как в случае одного компьютера K_0 , потому что сервером является K_0 компьютер. Особую роль в вычислительной системе играет функция распределения t классифицируемых объектов между n компьютерами, то есть функция $P(t, n)$. Функция $P(t, n)$ по мере того, как компьютеры закончили обработку одного объекта, передает им по заданному алгоритму новый классифицируемый объект для распознавания. Совокупность объектов для обработки на K_i -м компьютере обозначим как $X_i = \langle x_{i1}; x_{i2}; \dots \rangle$.

Суть работы такой вычислительной системы заключается в следующем: 1) сервер поочередно занимается предобработкой одного из t классифицируемых объектов (x_k -й объект) из файла $F_{\text{классиф.}}$ и передает его функции $P(t, n)$; 2) функция $P(t, n)$ распределяет полученный x_k -й объект на K_i -й компьютер согласно алгоритму реализации функции $P(t, n)$; 3) после распознавания на выходе K_i -го компьютера образуется результат в виде класса c_k ; 4) сервер, получив класс c_k , добавляет его в выходной файл с помощью функции $F_{\text{добав.}}(c_k)$.

Необходимо учитывать, что скорость распознавания (V_i средн. опред.) у компьютеров может быть различной, так как компьютеры могут обладать разной производительностью. Отсортируем компьютеры по скорости V_i средн. опред. в порядке возрастания. Если скорости V_i средн. опред. у нескольких компьютеров одинаковые, будем сортировать их в порядке возрастания индексов i (компьютеры представлены в виде K_i).

Введем обозначение K_i мин. произ. для отсортированного по возрастанию скорости V_i средн. опред. ряда

компьютеров, а их скорость V_i средн. опред. обозначим V_i мин. произ. средн. опред. Например, K_1 мин. произ. — первый компьютер в ряду, то есть компьютер, у которого скорость V_1 мин. произ. средн. опред. $\leq V_i$ мин. произ. средн. опред., где $1 < i \leq n$, то есть скорости V_i средн. опред. меньше, чем у него, среди n компьютеров нет.

Дополнительно отметим, что K_2 мин. произ. — второй компьютер в ряду, то есть, если из множества n компьютеров исключить K_1 мин. произ., то скорости V_i средн. опред. меньше V_2 мин. произ. средн. опред. среди множества $n-1$ компьютеров не будет.

Рассчитаем время распознавания файла $F_{\text{классиф.}}$ такой системой. Исследовав этапы работы параллельной вычислительной системы из n компьютеров, можно сделать вывод, что время распознавания будет зависеть от работы самого долгого по времени звена. Обозначим компьютер из этого звена как $K_{\text{долг. звена}}$, количество объектов, поступивших на $K_{\text{долг. звена}}$, как p долг. звена, скорость $V_{\text{средн. опред.}}$ компьютера $K_{\text{долг. звена}}$ как $V_{\text{долг. звена средн. опред.}}$. Стоит заметить, что компьютер $K_{\text{долг. звена}}$ может не совпадать с K_1 мин. произ.

Для расчетов времени распознавания также необходимо отметить следующее: 1) как говорилось выше, K_i -й компьютер получает объект для распознавания только после распознавания x_k -го объекта и записи сервером его класса в выходной файл; 2) из-за того, что чтение объектов из файла $F_{\text{классиф.}}$ происходит последовательно, а распознавание параллельно, может возникнуть ситуация ожидания чтения K_i -м компьютером своего объекта для распознавания, то есть K_i -й компьютер вынужден ждать, пока остальные компьютеры не прочитают свои объекты; 3) запись в выходной файл тоже происходит последовательно, что предполагает наличие возможности ожидания K_i -м компьютером своей очереди записи. Другими словами, необходима синхронизация процессов распознавания.

Заметим, что из-за синхронизации компьютер $K_{\text{долг. звена}}$ может оказаться последним в очереди на чтение и ему придется ждать, пока все $n-1$ компьютеры не прочитают свои объекты из файла $F_{\text{классиф.}}$. То же самое относится и к записи класса в

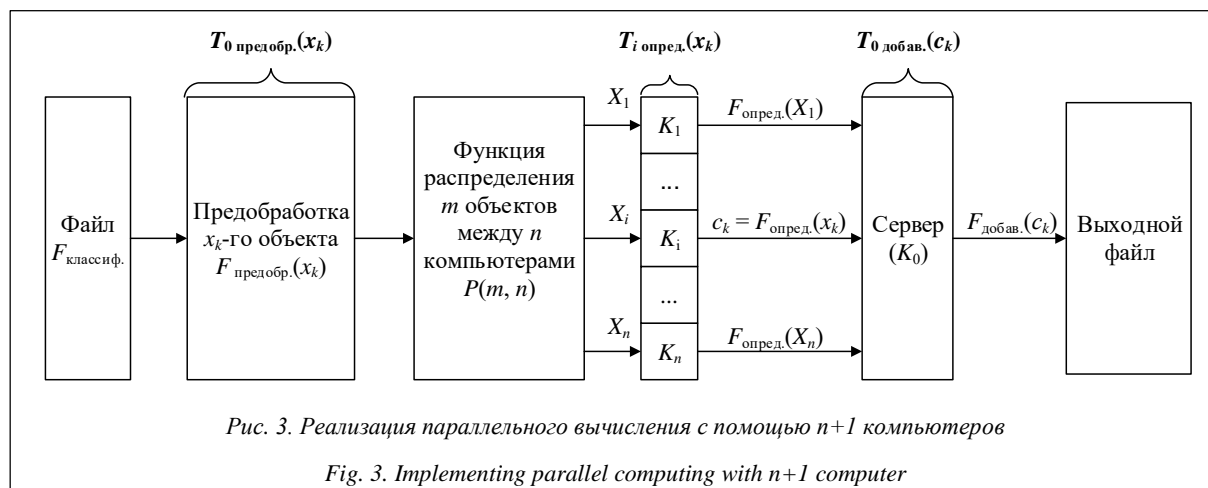


Рис. 3. Реализация параллельного вычисления с помощью $n+1$ компьютеров

Fig. 3. Implementing parallel computing with $n+1$ computer

выходной файл. С учетом этих условий рассчитаем время распознавания файла $F_{\text{классиф.}}$:

$$T_{\text{расп.}} = \left(\sum_{k=1}^n T_{0 \text{ предобр.}}(x_k) + \frac{1}{V_{\text{долг. звена средн. опред.}}} + \sum_{k=1}^n T_{0 \text{ добав.}}(c_k) \right) \times P_{\text{долг. звена}}, \quad (6)$$

где $\sum_{k=1}^n T_{0 \text{ предобр.}}(x_k)$ – сумма времени ожидания, пока все $n-1$ компьютеры не прочитают свои объекты, и времени чтения компьютером $K_{\text{долг. звена}}$ своего объекта; $\frac{1}{V_{\text{долг. звена средн. опред.}}}$ – время определения

класса одного классифицируемого объекта компьютером $K_{\text{долг. звена}}$; $\sum_{k=1}^n T_{0 \text{ добав.}}(c_k)$ – сумма времени ожидания, пока все $n-1$ компьютеры не запишут свои классы, и времени записи компьютером $K_{\text{долг. звена}}$ класса своего объекта; $P_{\text{долг. звена}}$ – количество объектов, которое распознает компьютер $K_{\text{долг. звена}}$.

Или, сгруппировав время по функциям, перепишем формулу (6):

$$T_{\text{расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{P_{\text{долг. звена}}}{V_{\text{долг. звена средн. опред.}}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (7)$$

Необходимо принять во внимание, что от реализации алгоритма распределения объектов функцией $P(m, n)$ будет зависеть и результат обработки, так как от функции $P(m, n)$ зависит число $P_{\text{долг. звена}}$. Рассмотрим несколько вариантов реализации этой функции.

Первый вариант. Функция $P(m, n)$ равномерно распределяет классифицируемые объекты между n компьютерами по следующему принципу: $X_i = \langle x_i; x_{i+n}; x_{i+2n}; \dots \rangle$.

Рассчитаем количество классифицируемых объектов, которые получит каждый компьютер. Для этого учтем тот факт, что m (количество объектов) может быть не кратным числу n (количество компьютеров). Тогда некоторые компьютеры получат меньше объектов, а другие больше объектов ($P_{\text{макс.}}$). Рассчитаем максимальное количество объектов, которые получают компьютеры при условии, что $m \geq n$:

$$P_{\text{макс.}} = \left\lceil \frac{m}{n} \right\rceil, \quad (8)$$

где $\left\lceil \frac{m}{n} \right\rceil$ – округление числа m/n до целого числа в большую сторону.

Чтобы воспользоваться формулой (7), необходимо определить $K_{\text{долг. звена}}$, а точнее, его скорость $V_{\text{долг. звена средн. опред.}}$ и количество объектов, которое он распознает. Из (8) и определения K_1 мин. произ.

можно сделать вывод, что $K_{\text{долг. звена}} = K_1$ мин. произ., $V_{\text{долг. звена средн. опред.}} = V_1$ мин. произ. средн. опред., а $P_{\text{долг. звена}} = P_{\text{макс.}}$. Тогда по формуле (7) рассчитаем время распознавания файла $F_{\text{классиф.}}$ при равномерном распределении объектов между n компьютерами:

$$T_{\text{равн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{P_{\text{макс.}}}{V_1 \text{ мин. произ. средн. опред.}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (9)$$

Проанализировав формулу (9), можно сделать вывод, что скорость V_1 мин. произ. средн. опред. обратно пропорциональна времени распознавания.

Выразим V_1 мин. произ. средн. опред. через V_0 средн. опред.: $V_1 \text{ мин. произ. средн. опред.} = \alpha * V_0 \text{ средн. опред.}$. Параметр α показывает, во сколько раз K_1 мин. произ. медленнее K_0 , если $0 < \alpha < 1$, и быстрее, если $\alpha > 1$.

Перепишем формулу (9) с учетом введенного параметра α :

$$T_{\text{равн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{P_{\text{макс.}}}{\alpha \times V_0 \text{ средн. опред.}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (10)$$

Рассмотрев формулу (10), можно заметить, что первое и третье слагаемые всегда будут вносить одинаковый вклад в общее время распознавания несмотря на изменение параметра α . Другими словами, как бы ни изменялся параметр α , сумма $\sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k)$ всегда будет нижней границей времени распознавания.

Исследуем влияние параметра α на время распознавания:

1) $\alpha = 1$, тогда $V_1 \text{ мин. произ. средн. опред.} = V_0 \text{ средн. опред.}$, а общее время распознавания, согласно формуле (10), является суммой времени распознавания $P_{\text{макс.}}$ объектов, времени предобработки m объектов и времени добавления результатов распознавания m объектов в выходной файл:

$$T_{\text{равн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{P_{\text{макс.}}}{V_0 \text{ средн. опред.}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k);$$

2) $\alpha = 2$, тогда $V_1 \text{ мин. произ. средн. опред.} = 2 * V_0 \text{ средн. опред.}$, а общее время распознавания рассчитывается следующим образом:

$$T_{\text{равн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{P_{\text{макс.}}}{2 \times V_0 \text{ средн. опред.}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k);$$

3) раз V_1 мин. произ. средн. опред. обратно пропорционально общему времени распознавания, определим α_{limit} , при котором $T_{0 \text{ расп.}} = T_{\text{равн. расп.}}$, то есть не будет выигрыша по времени при распознавании; приравняв формулы (5) и (10), определим $\alpha_{\text{limit}} = P_{\text{макс.}}/m$.

Рассмотрев формулу (10) и сравнив между собой время распознавания при различных значениях параметра α , можно сделать следующие выводы:

- если параметр $\alpha < p_{\text{макс.}}/m$, то выигрыша по времени при распознавании не будет;
- если параметр $\alpha = p_{\text{макс.}}/m$, то $T_0 \text{ расп.} = T_{\text{равн. расп.}}$;
- если $\alpha > p_{\text{макс.}}/m$, то можно будет наблюдать уменьшение времени распознавания.

Дополнительно отметим, что с ростом α время распознавания тоже будет уменьшаться, но только до своей нижней границы:

$$\sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k).$$

Рассмотрим достоинства и недостатки данной реализации функции $P(m, n)$. Среди достоинств можно выделить простоту реализации функции $P(m, n)$, отсутствие трудоемких вычислений и то, что заранее известно, какие объекты на какие компьютеры попадут.

Главным недостатком данной функции $P(m, n)$ является то, что она не учитывает особенности скорости обработки компьютерами объектов. Другими словами, у компьютеров может быть разная производительность, а время работы вычислительной системы n компьютеров определяется временем работы самым медленным K_1 мин. произ. компьютером. Отсюда вытекает возможность простоя вычислительных ресурсов других компьютеров.

Одним из способов решения проблемы простоя компьютеров является возврат системе компьютера, который выполнил свою работу, чтобы он мог выполнять работу уже другой задачи. Определить объем работы (количество объектов для распознавания) компьютера возможно благодаря тому, что заранее известно распределение объектов по компьютерам.

Но этот способ не поможет решить проблему, если задачу нужно решить в минимальные сроки, то есть критерием оптимальности выступит время распознавания. Несмотря на наличие свободных ресурсов, система не сможет ими воспользоваться для сокращения времени распознавания объектов.

Второй вариант. В предыдущем примере существует проблема простая компьютеров. Для ее решения можно предложить реализацию функции $P(m, n)$ на основе жадного алгоритма, суть которого заключается в том, чтобы отправлять объект для распознавания первому освободившемуся компьютеру, тем самым уменьшая время простоя.

Введем следующие параметры: число b – число классифицируемых объектов, которое пришло на вход K_1 мин. произ. компьютера; число $z = p_{\text{макс.}} - b$ – число перераспределенных объектов, то есть, если на вход K_1 мин. произ. компьютера поступит число b и $b < p_{\text{макс.}}$, то жадный алгоритм перераспределит z объектов с K_1 мин. произ. компьютера на другие компьютеры, тем самым уменьшая время их простоя.

В результате работы этого алгоритма могут сложиться две различные ситуации.

1. Жадному алгоритму не удалось переопределить число объектов, пришедших на вход K_1 мин. произ. компьютера, то есть $b = p_{\text{макс.}}$. Тогда $T_{\text{жадн. расп.}} = T_{\text{равн. расп.}}$ и нет никакого выигрыша во времени. Такая ситуация возможна, когда скорости V_i средн. опред. всех n компьютеров равны между собой. В таком случае распределение классифицируемых объектов между n компьютерами жадного алгоритма будет совпадать с распределением равномерного алгоритма.

2. Жадному алгоритму удалось переопределить число объектов, пришедших на вход K_1 мин. произ. компьютера, то есть $b < p_{\text{макс}}$. Тогда $T_{\text{ж.дн.}} \text{ расп.} < T_{\text{равни. расп.}}$. Для доказательства обратимся к рисунку 4.

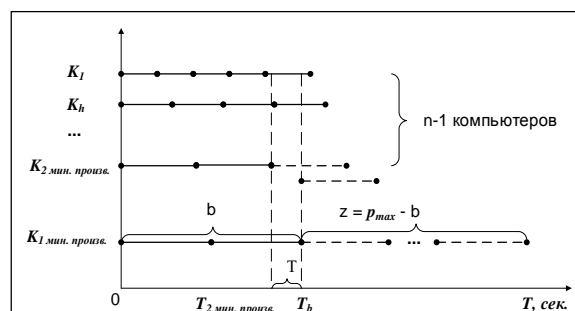


Рис. 4. Диаграмма производительности компьютеров
в работе жадного алгоритма

Fig. 4. Computer performance graph in the work of the greedy algorithm

Рассмотрим точку времени T_b – время, когда завершит свою работу K_1 мин. произ. компьютер по распознаванию b объектов. Стоит заметить, что в это время уже не будет объектов для распознавания. Иначе жадный алгоритм должен будет назначить K_1 мин. произ. компьютеру объект для распознавания, что противоречит тому, что на вход K_1 мин. произ. компьютера было подано только b объектов.

В точке времени T_b возможны два варианта работы жадного алгоритма.

1. Все остальные ($n-1$) компьютеры уже выполнили свою работу, то есть простаивают. Тогда время распознавания равно времени обработки K_1 мин. произ. компьютером b объектов:

$$T_{\text{ж.д.н. рас.}} = \sum_{k=1}^m T_0 \text{предобр.}(x_k) + \frac{b}{V_{1 \text{ мин. произв. средн. опед.}}} + \sum_{k=1}^m T_0 \text{добав.}(c_k). \quad (11)$$

2. Часть компьютеров из $n-1$ еще обрабатывают свои последние объекты. Определим компьютер с минимальной производительностью среди $n-1$ компьютеров, то есть компьютер K_2 мин. произ. Для простоты сдвинем время начала обработки последнего объекта K_2 мин. произ. компьютера с точки T_2 мин. произ. до точки T_b , то есть $T_b = T_2$ мин. произ. $+ \Delta T$.

Тогда время распознавания равно сумме времени обработки K_1 мин. произ. компьютером b объектов и времени обработки K_2 мин. произ. своего последнего объекта:

$$T_{\text{жадн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{b}{V_{1 \text{ мин. произв. средн. опред.}}} + \frac{1}{V_{2 \text{ мин. произв. средн. опред.}}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (12)$$

Очевидно, что время распознавания по формуле (12) больше времени распознавания по формуле (11). Поэтому дальше будем ориентироваться на время, рассчитанное по формуле (12).

Докажем теперь, что время распознавания по формуле (12) меньше времени распознавания по формуле (9). Очевидно, что можно опустить слагаемые $\sum_{k=1}^m T_{0 \text{ предобр.}}(x_k)$ и $\sum_{k=1}^m T_{0 \text{ добав.}}(c_k)$, так как они вносят одинаковый вклад в обе формулы. Для доказательства рассмотрим граничный случай, когда количество объектов в обеих формулах одинаково, то есть жадный алгоритм смог переопределить только один объект с компьютера K_1 мин. произ. Тогда формула (12) будет выглядеть следующим образом:

$$T_{\text{жадн. расп.}} = \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \frac{p_{\max} - 1}{V_{1 \text{ мин. произв. средн. опред.}}} + \frac{1}{V_{2 \text{ мин. произв. средн. опред.}}} + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k). \quad (13)$$

Проведя анализ формул (9) и (13), можно сделать вывод: для доказательства того, что $T_{\text{жадн. расп.}} < T_{\text{равн. расп.}}$, стоит сравнить между собой $1/V_{1 \text{ мин. произв. средн. опред.}}$ и $1/V_{2 \text{ мин. произв. средн. опред.}}$. Как отмечалось выше, $V_1 \text{ мин. произв. средн. опред.} \geq$

$\geq V_{2 \text{ мин. произв. средн. опред.}}$ и $T_b = T_{2 \text{ мин. произ.}} + \Delta T$. Это позволяет сделать вывод, что $1/V_{1 \text{ мин. произв. средн. опред.}} > 1/V_{2 \text{ мин. произв. средн. опред.}}$. Знак «строго» берется потому, что даже при равенстве скоростей между собой необходимо учитывать тот факт, что $T_b = T_{2 \text{ мин. произ.}} + \Delta T$, то есть K_2 мин. произ. компьютер начнет работу раньше, чем закончит работу K_1 мин. произ. компьютер.

В других же случаях с уменьшением параметра b будет уменьшаться $b/V_{1 \text{ мин. произв. средн. опред.}}$, что приводит к уменьшению времени распознавания.

Рассмотрим достоинства и недостатки реализации функции $P(m, n)$ в виде жадного алгоритма. Среди достоинств можно выделить простоту реализации функции $P(m, n)$, отсутствие трудоемких вычислений и возможность уменьшения времени простоя компьютеров, что отразится на времени распознавании.

Недостатки данной функции $P(m, n)$: неизвестно, какие объекты на какие компьютеры попадут; распределение по жадному алгоритму не всегда будет оптимальным, если критерием оптимальности выступит общее время распознавания.

Характеристики сети. Важнейшей характеристикой системы является время коммутации и передачи данных по каналам связи в сети, которое может оказать значительное влияние на длительность решения задачи. Фактически интерес представляет время, через которое после начала процесса вычисления пользователь получит его результат. Время получения результата на i -м компьютере (T_i получ. резул.) складывается из времени подключения (T_i подкл.) и времени на обработку данных (T_i обраб. дан.): $T_i \text{ получ. резул.} = T_i \text{ подкл.} + T_i \text{ обраб. дан.}$

Рассчитаем время, которое нужно потратить на этап обучения (рис. 5).

Суть обучения n компьютеров заключается в следующем: пользователь передает на сервер файл с обучающими объектами; сервер отправляет

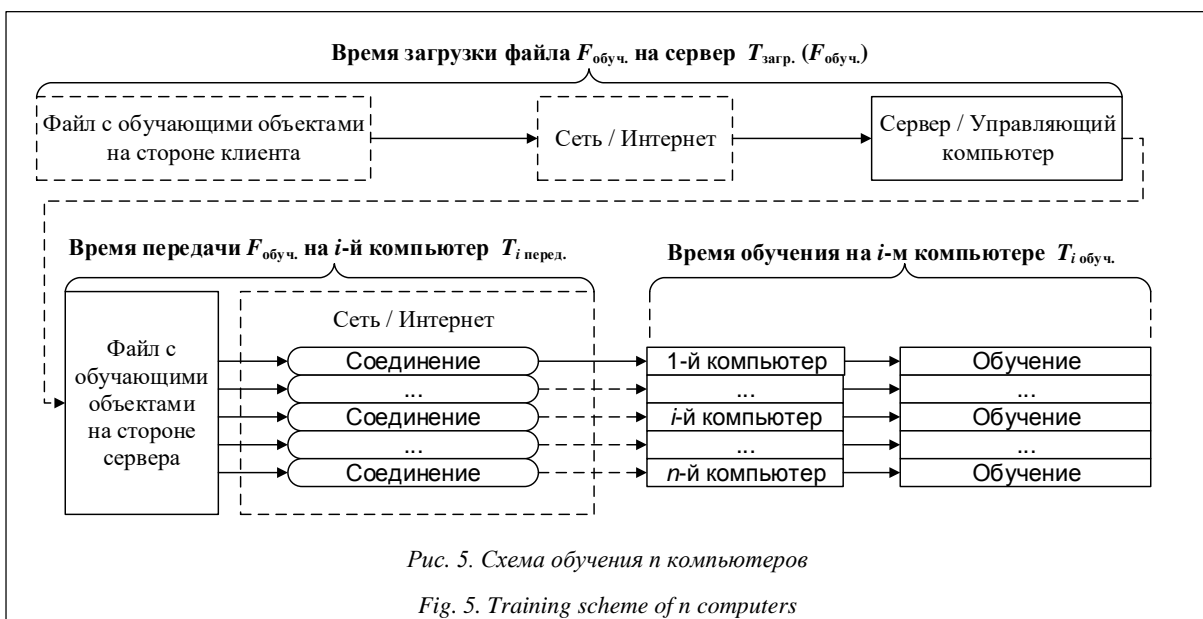


Рис. 5. Схема обучения n компьютеров

Fig. 5. Training scheme of n computers

далее этот файл параллельно n компьютерам; каждый компьютер обучается на этом файле. Время обучения n компьютеров рассчитывается следующим образом:

$$T_{\text{обуч.}} = T_{\text{загр.}}(F_{\text{обуч.}}) + T_{\text{макс. перед.}} + T_{\text{макс. обуч.}}, \quad (14)$$

где $T_{\text{загр.}}(F_{\text{обуч.}})$ – время загрузки файла $F_{\text{обуч.}}$ на сервер; $T_{\text{макс. перед.}}$ – максимальное время передачи файла $F_{\text{обуч.}}$ на один из n компьютеров; $T_{\text{макс. обуч.}}$ – максимальное время обучения одного из n компьютеров на основе файла $F_{\text{обуч.}}$.

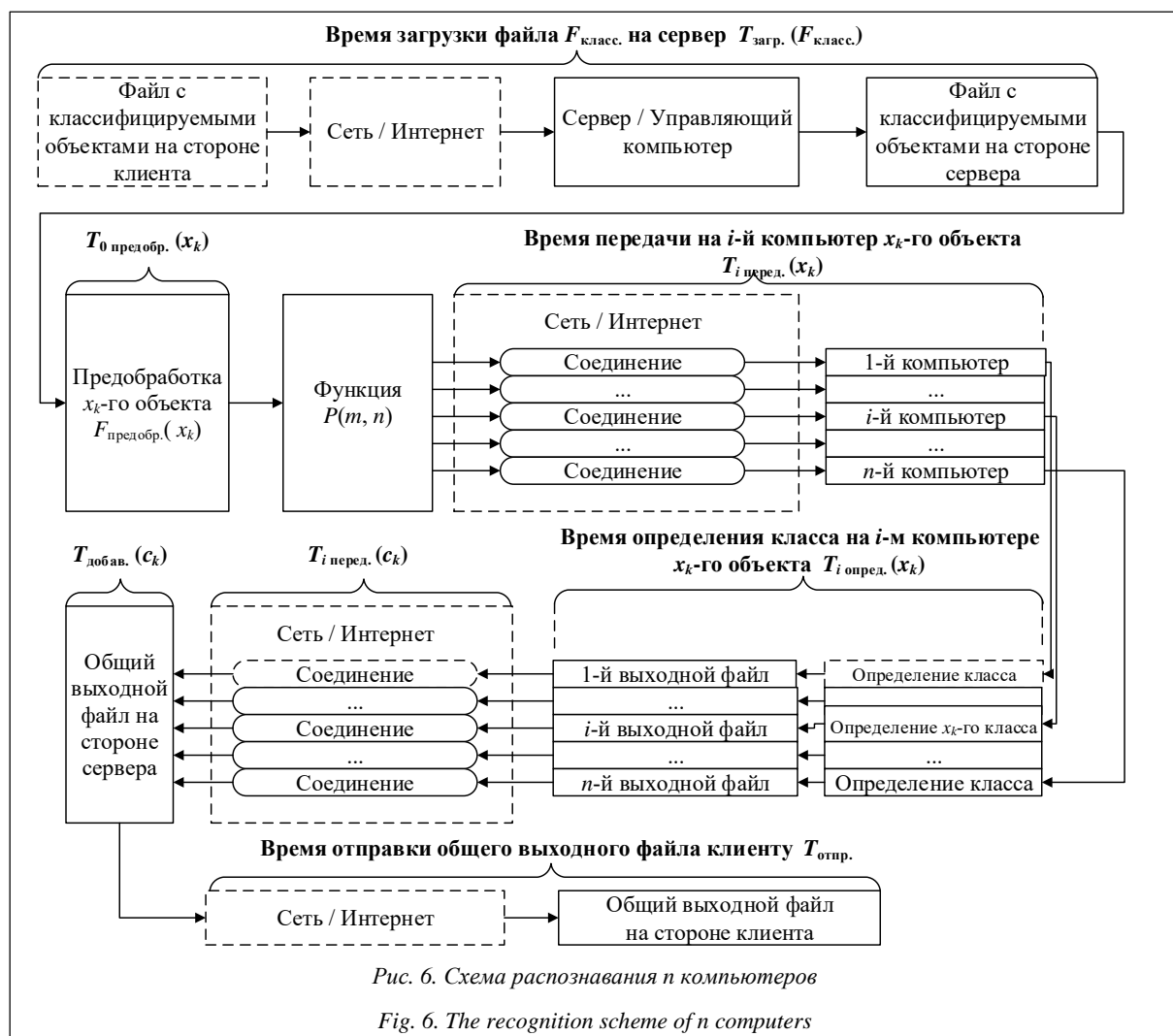
Рассчитаем время, необходимое на распознавание (рис. 6).

Распознавание происходит следующим образом: 1) пользователь передает на сервер файл с классифицируемыми объектами; 2) сервер осуществляет предобработку и распределение классифицируемых объектов из этого файла параллельно n компьютерам согласно реализации функции $P(m, n)$; 3) i -й компьютер, получив классифицируемый объект, начинает процесс определения его класса; 4) класс распознанного объекта записывается в выходной файл i -го компьютера, после чего отправляется на сервер; 5) сервер после получения

класса объекта начинает добавление его в общий выходной файл; 6) после добавления всех классов в общий выходной файл сервер отправляет общий выходной файл клиенту.

Как уже отмечалось, общее время работы системы зависит от времени работы звена с максимальным временем обработки своих объектов, то есть компьютером $K_{\text{долг. звена}}$. Для равномерного распределения компьютер $K_{\text{долг. звена}}$ определяется параметром V_1 мин. произ. средн. опред., но при этом не учитывались характеристики канала связи. Поэтому теперь для K_i -го компьютера время расчета определения класса одного объекта складывается из следующих параметров: $T_{i \text{ сетев. опред.}} = T_{i \text{ перед.}}(x_k) + T_{i \text{ сред. опред.}}(x_k) + T_{i \text{ перед.}}(c_k)$, где $T_{i \text{ перед.}}(x_k)$ – время передачи x_k -го объекта с сервера на i -й компьютер; $T_{i \text{ сред. опред.}}(x_k)$ – время определения класса x_k -го объекта на i -м компьютере; $T_{i \text{ перед.}}(c_k)$ – время передачи класса x_k -го объекта с i -го компьютера на сервер.

Другими словами, компьютер $K_{\text{долг. звена}}$ будет определяться не только производительностью компьютера, но и характеристиками канала связи. Те-



перь с учетом вышесказанных замечаний и формулы (7) рассчитаем общее время распознавания:

$$T_{\text{расп.}} = T_{\text{загр.}}(F_{\text{классиф.}}) + \sum_{k=1}^m T_{0 \text{ предобр.}}(x_k) + \\ + \sum_{k=1}^{P_{\text{долг. звена}}} T_{\text{долг. звена перед.}}(x_k) + \frac{P_{\text{долг. звена}}}{V_{\text{долг. звена средн. опред.}}} + \\ + \sum_{k=1}^{P_{\text{долг. звена}}} T_{\text{долг. звена перед.}}(c_k) + \sum_{k=1}^m T_{0 \text{ добав.}}(c_k) + T_{\text{отпр.}}, \quad (15)$$

где $T_{\text{загр.}}(F_{\text{классиф.}})$ – время загрузки файла $F_{\text{классиф.}}$ на сервер; $\sum_{k=1}^{P_{\text{долг. звена}}} T_{\text{долг. звена перед.}}(x_k)$ – время передачи

$P_{\text{долг. звена}}$ объектов на компьютер $K_{\text{долг. звена}}$; $\sum_{k=1}^{P_{\text{долг. звена}}} T_{\text{долг. звена перед.}}(c_k)$ – время передачи классов $P_{\text{долг. звена}}$ объектов на сервер; $T_{\text{отпр.}}$ – время отправки общего выходного файла клиенту.

Если время решения задачи $T_{0 \text{ реш. зад.}} = T_{0 \text{ обуч.}} + T_{0 \text{ расп.}}$ на K_0 компьютере меньше времени $T_{\text{паралл. реш. зад.}} = T_{\text{паралл. обуч.}} + T_{\text{паралл. расп.}}$ (с учетом времени коммутации и передачи данных по сети), то распараллеливание не повысит общую производительность системы. Здесь $T_{0 \text{ обуч.}}$ рассчитывается по формуле (1), $T_{0 \text{ расп.}}$ – по формуле (3), $T_{\text{паралл. обуч.}}$ – по формуле (14), $T_{\text{паралл. расп.}}$ – по формуле (15). Такая ситуация может возникнуть при значительном влиянии характеристик канала связи, в том числе времени коммутации и передачи данных.

Минимизировать $T_{i \text{ обраб. дан.}}$ можно за счет установки высокопроизводительного оборудования. Минимизация $T_{\text{расп.}}$ достигается путем установки дополнительных компьютеров, тем самым снижая среднее значение $T_{i \text{ обраб. дан.}}$, а также стремлением к максимально низкому значению $T_{i \text{ подкл.}}$, размещая сервер и компьютеры в одной локальной сети. Следует помнить, что при $T_{i \text{ обраб. дан.}}$, близком к $T_{i \text{ подкл.}}$, требуется решить задачу выбора оптимального количества компьютеров, дабы избежать отрицательного эффекта от применения системы. Оптимизировать параметры $T_{i \text{ подкл.}}$ и $T_{i \text{ обраб. дан.}}$ стоит исходя из соображений целесообразности и общей эффективности.

В ходе исследования было разработано web-ориентированное инструментальное средство по распознаванию образов в идее расширяемой библиотеки методов на основе прецедентов, способное не только работать на одном сервере, но и распараллеливать обработку данных между компьютерами.

Реализация такой параллельности была представлена в статье [12], где описаны теоретические и экспериментальные разработки двух схем распараллеливания на основе grid-технологий: с распараллеливанием алгоритма обучения и с распараллеливанием алгоритма распознавания, а также представлены элементы практических разработок различных этапов параллельного распознавания на

n компьютерах на основе метода k ближайших соседей.

Перспективность исследования определяется практической эффективностью разработанного web-приложения. Практика показывает, что распараллеливание позволяет эффективно применять вычислительные ресурсы для решения задач в минимальные сроки, а применение web-технологии – повысить объем обрабатываемых данных и скорость доступа к ним.

Теоретические выкладки и расчеты работы являются базовыми для практических разработок данного web-приложения. В статье обозначены ограничения на применение вычислительных ресурсов, несоблюдение которых приводит к ухудшению результатов времени распознавания. Также рассчитана минимальная граница времени распознавания вне зависимости от количества используемых компьютеров.

Реализуемая архитектура инструментария позволяет решать задачу повышения эффективности вычислительных ресурсов через процедуры настройки структуры подключаемых методов и алгоритмов обработки данных и реконфигурирования подключаемых технических вычислительных средств Internet-сети.

Литература

1. Козлов А.Н. Интеллектуальные информационные системы. Пермь: Изд-во Перм. ГСХА, 2013. 278 с.
2. Громов Ю.Ю., Иванова О.Г., Алексеев В.В. [и др.]. Интеллектуальные информационные системы и технологии. Тамбов: Изд-во ТГТУ, 2013. 244 с.
3. Han J., Kamber M., Pei J. Data mining: concepts and techniques, Elsevier, 2012, 744 p.
4. Фомин Я.А. Распознавание образов: теория и применения. М.: Фазис, 2014. 460 с.
5. Мерков А.Б. Распознавание образов: введение в методы статистического обучения. М.: Едиториал УРСС, 2011. 254 с.
6. Sandoval J., Hernández G. Learning of natural trading strategies on foreign exchange high-frequency market data using dynamic bayesian networks. Machine Learning and Data Mining in Pattern Recognition. Proc. 10th Intern. Conf., 2014, pp. 408–421.
7. Сикудер Д.В., Фомин В.В. Проектные решения web-ресурса интеллектуального анализа данных и поддержки принятия решений на основе унифицированного класса методов решающих функций. СПб: Изд-во РГПУ, 2016. 102 с.
8. Марманис Х., Бабенко Д. Алгоритмы интеллектуального Интернета. Передовые методики сбора, анализа и обработки данных; [пер. с англ.]. СПб: Символ-Плюс, 2011. 466 с.
9. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 1. С. 45–57.
10. Петров Е.В., Фомин В.В. Organization of parallel processing for implementation of web data mining system // Ученые записки РГТУ. 2014. № 33. С. 149–154.
11. Алпатов А.Н. Развитие распределенных технологий и систем // Перспективы науки и образования. 2015. № 2. С. 60–66.
12. Фомин В.В., Александров И.В. Об одном опыте применения параллельных вычислений при разработке web-инструментария распознавания образов // Программные продукты, системы и алгоритмы. 2017. № 1. URL: <http://swsys-web.ru/experience-in-parallel-computing-web-tool-for-pattern-recognition.html> (дата обращения: 02.04.2017).

PARALLEL COMPUTING WHEN IMPLEMENTING WEB-BASED PATTERN RECOGNITION TOOLS BASED ON USE CASE METHODS

V.V. Fomin¹, Dr.Sc. (Engineering), Professor, v_v_fomin@mail.ru

I.V. Aleksandrov, Postgraduate Student, chrono555@yandex.ru

¹ Herzen State Pedagogical University of Russia, Reki Moyki Quay 48, St. Petersburg, 191186, Russian Federation

Abstract. The article considers a software solution aimed at improving image recognition quality and effectiveness of machine learning tools using grid technologies. The paper formulates strategic directions for pattern recognition tools as a software system based on the principles of distributed systems, parallelism and adaptive configuration of computing resources. The article considers the structure of web tools for pattern recognition using the concept of the algorithm library. It also describes the algorithmic solutions for parallelizing learning and recognition algorithms on the basis of classical data mining algorithms that are proved themselves in practice. Such algorithms include precedent (case-based reasoning) methods or methods based on proximity metrics. They have a great potential to parallelize computational processes and develop parallel algorithms for their implementation.

The search for ways to improve the productivity of computers, especially when implementing web tools based on resource-intensive computer algorithms for machine recognition and prediction, led to the decision to create a grid system. The architecture and implementation of the grid system considered in the article assumes parallelization and organization of distributed computations on a multi-machine basis using Internet technologies. It allows obtaining similar computing power as multiprocessor computer systems but the cost is much lower.

The article considers the problem of increasing the efficiency of computing resources with the possibility of reconfiguring the structure of Internet connections, including the procedure of configuring a computer network structure, connectable communication channels and dedicated servers depending on the original algorithms and data. The paper presents dependencies of operation execution time parameters on the service discipline that adapts the system to user requests. Here the tasks are ranked according to resource intensity and computing power corresponding to their rank is allocated to them.

Keywords: intelligent information systems, pattern recognition, machine learning, web systems, parallel computing.

References

1. Kozlov A.N. *Intellektualnye informatsionnye sistemy* [Intelligent Information Systems]. Perm, Permskaya GSKhA Publ., 2013, 278 p.
2. Gromov Yu.Yu., Ivanova O.G., Alekseev V.V. *Intellektualnye informatsionnye sistemy i tekhnologii* [Intelligent Information Systems and Technologies]. Tambov, TGTU Publ., 2013, 244 p.
3. Han J., Kamber M., Pei J. *Data mining: concepts and techniques*. Elsevier Publ., 2012, 744 p.
4. Fomin Ya.A. *Raspoznavanie obrazov: teoriya i primeneniya* [Image Identification: Theory and Applications]. Moscow, Fazis Publ., 2014, 460 p.
5. Merkov A.B. *Raspoznavanie obrazov: vvedenie v metody statisticheskogo obucheniya* [Image Identification: Introduction and Statistical Learning Methods]. Moscow, Editorial URSS Publ., 2011, 254 p.
6. Sandoval J., Hernández G. Learning of Natural Trading Strategies on Foreign Exchange High-Frequency Market Data Using Dynamic Bayesian Networks. *Machine Learning and Data Mining in Pattern Recognition: Proc. 10th Int. Conf.* St. Petersburg, Russia, Springer Publ., 2014, pp. 408–421.
7. Sikuler D.V., Fomin V.V. *Proektnye resheniya web-resursa intellektualnogo analiza dannykh i podderzhki prinyatiya resheny na osnove unifikirovannogo klassa metodov reshayushchikh funktsy* [Project Decisions of a Web Source of Data Mining and Decision Support Based on a Decision Function Method Unified Class]. St. Petersburg, RGPU Publ., 2016, 102 p.
8. Marmanis Kh., Babenko D. *Algorithms of the Intelligent Web*. Manning Publ., 2009, 368 p. (Russ.ed.: St. Petersburg, Simvol-Plyus Publ., 2011, 466 p.).
9. Varshavsky P.R., Ereemeev A.P. Modeling Case-Based Reasoning in Intelligent Decision Support Systems. *Iskusstvenny intellekt i prinyatie resheny* [Artificial Intelligence and Decision Making]. 2009, no. 1, pp. 45–57 (in Russ.).
10. Petrov E.V., Fomin V.V. Organization of parallel processing for implementation of web data mining system. *Uchenye zapiski Rossiyskogo gos. gidrometeorologicheskogo universiteta: nauchno-teoretichesky zhurnal* [RSHU Proc. Jour.]. 2014, no. 33, pp. 149–154 (in Russ.).
11. Alpatov A.N. Problems of distributed systems. *Perspektivy nauki i obrazovaniya* [Perspectives of Science and Education]. 2015, no. 2 (14), pp. 60–66 (in Russ.).
12. Fomin V.V., Aleksandrov I.V. On one operational experience of parallel computing when developing image identification web tools. *Programmnye produkty, sistemy i algoritmy* [Software Products, Systems and Algorithms]. 2017, no. 1. (in Russ.). Available at: <http://swsys-web.ru/experience-in-parallel-computing-web-tool-for-pattern-recognition.html> (accessed April 2, 2017).

УДК 621.3.06
DOI: 10.15827/0236-235X.118.217-226

Дата подачи статьи: 10.04.17
2017. Т. 30. № 2. С. 217–226

ОДНОРОДНАЯ РАСПРЕДЕЛИТЕЛЬНАЯ ЗАДАЧА НА ОСНОВЕ МОДЕЛЕЙ АДАПТИВНОГО ПОВЕДЕНИЯ МУРАВЬИНОЙ КОЛОНИИ

Б.К. Лебедев, д.т.н., профессор, lebedev.b.k@gmail.com;

О.Б. Лебедев, к.т.н., доцент, lebedev.ob@mail.ru;

Е.М. Лебедева, аспирант, lebedeva.el.m@mail.ru

*(Институт компьютерных технологий и информационной безопасности
Южного федерального университета, ул. Энгельса, 1, г. Таганрог, 347928, Россия)*

В данной работе предлагается решение однородной распределительной задачи. Приводится постановка этой задачи, рассматриваются основные группы алгоритмов ее решения – приближенные и точные, а также их достоинства и недостатки. Описана предлагаемая новая парадигма комбинаторной оптимизации, базирующаяся на моделировании адаптивного поведения муравьиной колонии.

Решением однородной распределительной задачи является ее графическое представление в виде двудольного графа. Для решения данных задач были предложены новые механизмы. Основу метаэвристики алгоритма на основе муравьиной колонии составляет комбинация двух техник. Базовая техника состоит в поиске наилучшего решения с использованием механизмов адаптивного поведения муравьиной колонии. Муравей строит какое-то конкретное решение, при этом используется встроенная процедура, в основе которой лежит конструктивный алгоритм. Построенный на графе поиска решений двудольный граф – основное отличие предлагаемого муравьиного алгоритма от существующей канонической парадигмы.

При нахождении оптимальных решений оптимизационных задач, которые допускают представление решений в виде двудольных графов, данный подход будет достаточно эффективным.

Проведенные исследования показали, что муравьиный алгоритм позволяет получать более качественные решения, чем известные алгоритмы. Сравнив результаты, можно сказать, что они улучшились на 3–4 %.

Ключевые слова: задача о назначениях, двудольный граф, оптимизация, роевой интеллект, муравьиная колония, адаптивное поведение, однородная распределительная задача.

В области теории расписаний самым распространенным направлением является исследование классических однородных распределительных задач [1]. Такие задачи часто применяются в различных сферах человеческой деятельности, например там, где необходимо эффективно выполнить организацию и планирование каких-либо работ, различных заданий и требований. В общем случае имеем какое-то количество работ и какое-то количество исполнителей. Выполнение любым работником какой-либо (но только одной) работы происходит с разными (неодинаковыми) затратами. Необходимо распределить работы таким образом, чтобы обойтись минимальными затратами. Примером распределительной задачи является задача составления плана выполнения комплекса программ в многопроцессорных вычислительных системах [2]. Она является минимаксной однородной распределительной задачей теории расписаний.

Поскольку распределительная задача относится к классу NP-полных задач, разработка новых, более эффективных методов, используемых для их решения, является актуальной проблемой теории расписаний. Зарубежными и российскими учеными разработано множество алгоритмов и методов, отличающихся как эксплуатационными свойствами, так и областью применения и решающих однородные распределительные задачи. Все методы решения однородных распределительных задач делятся на две группы – приближенные и точные.

Самыми известными и наиболее эффективными из методов точного решения распределительной задачи являются алгоритмы, в основе которых лежат принципы метода ветвей и границ [2–4]. Однако следует отметить, что эти алгоритмы имеют свойство экспоненциального роста сложности искомого решения относительно размерности распределительной задачи. Это особенно заметно, когда распределяемые в задаче задания характеризуются небольшой вариацией значений их размеров. Основные два дефекта, которые часто возникают в алгоритмах на основе метода ветвей и границ, указаны в работе [4]. В ней предлагается модификация, которая может уменьшить комбинационную сложность алгоритма, предложенного Романовским [2]. Главная особенность метода ветвей и границ состоит в том, что обязательно возникнет такой пример, при решении которого необходим полный перебор. Поэтому для задач повышенной размерности возникнут большие временные затраты, хотя комбинаторная сложность уменьшится.

Алгоритмы приближенного поиска не могут гарантированно найти при решении распределительной задачи глобальный оптимум. Они позволяют получать некоторое допустимое или приемлемое решение [4–6]. Точность алгоритмов этого класса невысока, но основное достоинство заключается в высокой скорости получения решения, которая имеет полиномиальную, а иногда даже линейную

зависимость от заданного порядка задачи. Наибольшее распространение в классе приближенных алгоритмов получили списочные и эвристические методы. В основном списочные методы имеют линейную сложность решения (относительно размерности задачи). В связи с этим, с точки зрения времени построения расписаний, они обладают большой эффективностью. К сожалению, при получении решений погрешность может достигнуть 30 %, что неприемлемо к заданным практическим требованиям. Интерес к использованию эвристических подходов возник в связи с низкой точностью приближенных списочных алгоритмов и высокими вычислительными требованиями точных методов.

В литературе [5–11] приведены разработанные ранее алгоритмы (последовательные, итерационные и т.д.). Их обзор, сравнение и анализ показали, что для разработки более эффективного алгоритма требуются новые технологии и подходы. Возможности применения генетического алгоритма для решения однородной распределительной задачи рассматриваются в [12, 13]. Как показывают исследования, алгоритм лучше срабатывает при значительном увеличении популяции и числа итераций. В [14] приведены некоторые возможные модификации использования алгоритма поиска Табу. Эксперименты показывают, что этот алгоритм не всегда способен дать оптимальные решения. Качество решений возрастает в том случае, если его использовать совместно с алгоритмом случайного поиска. В [15] авторами было проведено исследование возможности использования нейронных сетей для решения однородной распределительной задачи и приведены результаты поставленных экспериментов. Предлагается использовать динамическую нейронную сеть, основанную на нейронах непрерывного и двоичного типов. Из результатов экспериментальных исследований следует, что метод позволяет получать решения, результаты которых могут быть улучшены.

Как показывают эксперименты, приближенные (эвристические) методы могут давать большую погрешность, неприемлемую в большинстве случаев. Возникает необходимость в методах, сочетающих противоречивые свойства: полиномиальная зависимость времени решения задачи от ее размерности и точность, близкая к оптимальной. Таким образом, работа посвящается актуальной научно-технической проблеме: исследованию приближенных методов решения задачи распределения программ в многопроцессорных системах с целью повышения точности их работы.

В настоящее время наиболее перспективным является интенсивно разрабатываемое в последние годы научное направление, объединяющее различные математические методы, в которых принятие решений осуществляется на основе принципов, содержащих природные механизмы [15–21]. Эти методы являются итерационными, эвристическими

методами случайного поиска. Наиболее активно развиваются методы роевого интеллекта (Swarm-Intelligence) [17–21]. В них совокупность простых агентов конструирует стратегию своего поведения без глобального управления.

Одним из новейших и наиболее перспективных мультиагентных методов интеллектуальной оптимизации является метод муравьиной колонии [16, 17]. В работе представлен разработанный алгоритм для решения однородной распределительной задачи, использующий модель адаптивного поведения муравьиной колонии.

Постановка однородной распределительной задачи теории расписаний

Пусть исполнительная система состоит из n идентичных, параллельно работающих исполнителей $E = \{e_i / i = 1, 2, \dots, n\}$. На вход системы поступает множество m независимых заданий (работ) $W = \{w_j / j = 1, 2, \dots, m\}$, которые необходимо распределить между исполнителями. Известна стоимость (ресурс выполнения) каждого j -го задания r_j , и она одинакова для любого i -го исполнителя e_i . Таким образом, множеству W сопоставлено множество стоимостей: $R = \{r_j / j = 1, \dots, m\}$. l -м решением однородной распределительной задачи является множество $V^l = \{W_i^l / i = 1, 2, \dots, n\}$, в котором подмножества заданий W_i^l отвечают обязательному свойству:

$$(\forall i, j) [((i \neq j) \& (W_i^l \in V^l) \& (W_j^l \in V^l)) \rightarrow ((W_i^l \cap W_j^l = \emptyset) \& (\cup W_i^l = W))]. \quad (1)$$

Запланированная вариантом решения V^l загрузка заданиями каждого исполнителя e_i оценивается ресурсом:

$$R_i^l = \sum_k r_k, \quad (2)$$

где $r_k / w_k \in W_i^l$.

В качестве оценки решения V^l рассматривается величина

$$F^l = \max(R_i^l). \quad (3)$$

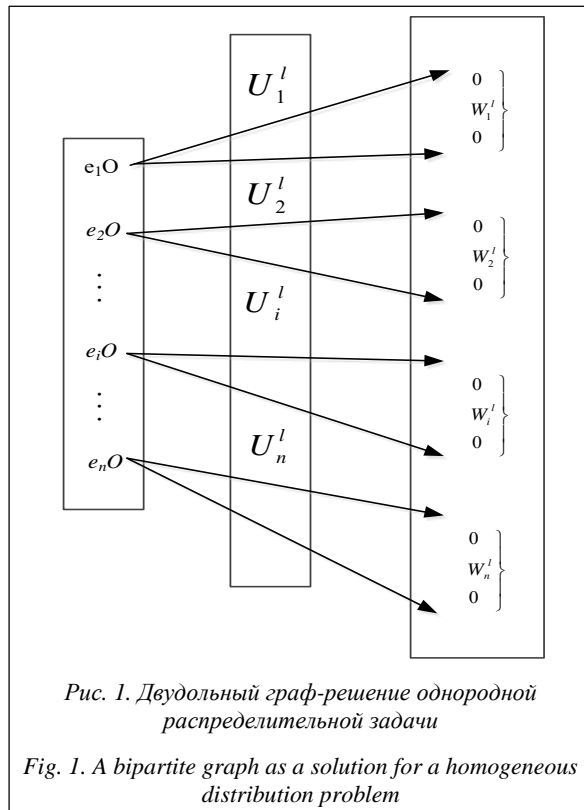
Наиболее распространенным критерием оптимизации однородной распределительной задачи является минимаксный критерий

$$F = \min F^l = \min \max (R_i^l). \quad (4)$$

Это связано с тем, что в большинстве реальных распределительных задач наиболее часто возникает задача минимизации времени выполнения всего комплекса заданий на заданной исполнительной системе. Чем быстрее исполнительная система обслуживает поступающие на выполнение задания, тем выше ее производительность, а значит, выше и экономическая эффективность предприятия, использующего данную систему.

Представим решение распределительной задачи в виде двудольного графа: $H^l = (E \cup W, U^l)$, где $E = \{e_i / i = 1, 2, \dots, n\}$ – множество вершин (первая доля), соответствующих множеству исполнителей, а $W = \{w_j / j = 1, 2, \dots, m\}$ – множество вершин (вторая доля), соответствующих множеству заданий (ра-

бот); U^l – множество ребер, связывающих вершины множества E с вершинами множества W . Для наглядности представления конкретного решения V^l сгруппируем множество заданий W в подмножества (рис. 1).



Пусть U_i^l – множество ребер двудольного графа H^l , связывающих вершину e_i с вершинами множества W_i^l , $\cup U_i^l = U^l$. Отличительной особенностью представленного двудольного графа $H^l = (E \cup W, U^l)$ является то, что число ребер множества U_i^l равно числу вершин множества W_i^l . Каждое ребро $u_i \in U_i^l$, с одной стороны, инцидентно вершине e_i , а с другой – инцидентно только одной вершине $w_k \in W_i^l$. Назовем двудольный граф, представляющий решение V^l , графом-решением H^l . Отметим, что локальная степень любой вершины $w_j \in W$ равна единице, а локальная степень вершины e_i равна мощности множества W_i^l , то есть $\rho(w_j) = 1$, $\rho(e_i) = |W_i^l|$.

В работе поиск решения V^l сводится к поиску на полном двудольном графе H_{nm} такого графа-решения H^l , для которого оценка F^l имеет минимальное значение.

Решение однородной распределительной задачи на основе моделей адаптивного поведения муравьиной колонии

Предлагаемая для муравьиного алгоритма метаэвристика состоит из комбинации двух техник. Сначала на базовом методе строится общая схема. Далее в построенную схему встраивается та или

иная процедура. Следует отметить, что встроенная процедура – это практически всегда самостоятельный алгоритм решения той же задачи, что и метаэвристический метод в целом. Базовый метод заключается в реализации итерационной процедуры поиска лучшего решения на основе механизмов адаптивного поведения муравьиной колонии. Основу встроенной процедуры составляет конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения. В оптимизации муравьиными колониями [8] конструктивный блок (деятельность искусственных муравьев) играет ключевую роль. В нашем случае в качестве интерпретации решения однородной распределительной задачи служит двудольный граф-решение H^l .

Рассмотрим принципы решения однородной распределительной задачи методами муравьиной колонии. Поиск решений осуществляется на полном двудольном графе. Как указывалось выше, базовый метод заключается в реализации итерационной процедуры поиска лучшего решения. Работа поисковой процедуры начинается с построения в соответствии со спецификой решаемой задачи графа поиска решений. Для поиска решений формируется полный двудольный граф $H_{nm} = (E \cup W, U)$. В графе H_{nm} каждая вершина e_i связана со всеми вершинами множества W , а каждая вершина w_j связана со всеми вершинами множества E , то есть $\rho(w_j) = |U|$, $\rho(e_i) = |W|$, $|U| = nm$. Задается размер популяции искусственных муравьев. За каждым муравьем закрепляется стартовая вершина. В качестве стартовых вершин рассматриваются вершины $w_j \in W$.

Моделирование поведения муравьев связано с распределением феромона на ребрах графа H_{nm} . На начальном этапе на всех ребрах U графа H_{nm} откладывается одинаковое (небольшое) количество феромона Q/v , где $v = |U|$. Параметр Q задается априори. Будем обозначать граф H_{nm} после отложения на нем на итерации t феромона как $H_{nm}(t)$. После начального отложения – $H_{nm}(0)$. Процесс поиска решений итерационный. Каждая итерация t включает три этапа. На первом этапе каждой итерации t выполняются процедуры муравьиного алгоритма. Каждый l -й агент формирует на ребрах графа $H_{nm}(t-1)$ свой собственный граф-решение $H^l(t)$, определяется решение $V^l(t)$, соответствующее графу-решению $H^l(t)$, и оценка решения $F^l(t)$.

На втором этапе итерации t каждый муравей откладывает феромон на ребрах графа $H_{nm}(t-1)$, соответствующих ребрам построенного графа-решения $H^l(t)$.

Количество феромона $\Delta\tau_i(t)$, откладываемое муравьем a_l на каждом ребре построенного графа-решения $H^l(t)$, определяется следующим образом: $\Delta\tau_i(t) = Q / F^l(t)$, где t – номер итерации; Q – общее количество феромона, откладываемое муравьем на ребрах графа-решения $H^l(t)$; $F^l(t)$ – целевая функция для решения, полученного муравьем a_l на t -й

итерации. Чем меньше $F^l(t)$, тем больше феромона откладывается на ребрах построенного графа-решения $H^l(t)$ и, следовательно, тем больше вероятность выбора этих ребер при построении графа-решения на следующей итерации.

Обозначим $f_{ij}(t)$ суммарное количество феромона, скопившееся на ребре u_k двудольного графа $H_{nm}(t)$, связывающего вершину e_i с вершиной w_j после выполнения второго этапа итерации t .

После того как каждый агент сформировал решение и отложил феромон, на третьем этапе итерации t происходит общее испарение феромона на ребрах двудольного графа $H_{nm}(t)$ в соответствии с формулой $f_{ij}(t) = f_{ij}(t)(1 - \rho)$, где ρ – коэффициент обновления.

После выполнения всех действий на итерации t находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Рассмотрим теперь конструктивный алгоритм построения муравьем двудольного графа-решения $H^l(t)$ на полном двудольном графе $H_{nm}(t-1)$.

Последовательно (или случайным образом) выбираются вершины $w_j \in W$ двудольного графа $H_{nm}(t-1)$, начиная с первой (стартовой). Для выбранной вершины w_j определяется набор ребер $U_j(t)$, связывающих w_j со всеми вершинами e_i множества E . Для каждого ребра $u_k \in U_j(t)$, связывающего вершину e_i с вершиной w_j , определяется параметр f_{ij} – суммарный уровень феромона на этом ребре.

Вероятность P_k включения ребра $u_k \in U_j(t)$ в формируемый граф-решение $H^l(t)$ определяется соотношением $P_k = f_{ij} / \sum_i (f_{ij})$.

Агент с вероятностью P_k выбирает одно из ребер, которое включается в формируемый граф-решение $H^l(t)$.

Временная сложность этого алгоритма зависит от времени жизни колонии t (число итераций), количества исполнителей n и числа работ m и определяется как $O(l \cdot c^2 \cdot m)$. Далее приведены алгоритм поведения муравьиной колонии и алгоритм муравья.

Алгоритм поведения муравьиной колонии.

1. Задаются число исполнителей – n , число работ – m , начальное количество феромона – Q .

2. Строится полный двудольный граф $H_{nm}(0) = (E \cup W, U(0))$, на ребрах которого отложено начальное количество феромона.

3. За каждым муравьем a_l закрепляется стартовая вершина $w_l \in W$.

4. Задаются число итераций – NT , число муравьев, формирующих независимо друг от друга решения на одной итерации, – NR .

5. $t=1$ (t – номер итерации).

6. $l=1$ (l – номер агента).

7. (Алгоритм муравья). Муравей a_l строит на полном двудольном графе $H_{nm}(t-1) = (E \cup W, U(t-1))$ двудольный граф-решение $H^l(t)$.

8. Рассчитывается оценка $F^l(t)$ построенного двудольного графа-решения $H^l(t)$.

9. Если $l < NR$, то $l = l + 1$ и переход к пункту 7, иначе переход к пункту 10.

10. $l=1$.

11. Муравей a_l откладывает на ребрах полного двудольного графа $H_{nm}(t-1)$, соответствующих ребрам построенного двудольного графа-решения $H^l(t)$, феромон в количестве $\Delta\tau_l(t) = Q / F^l(t)$.

12. Если $l < NR$, то $l = l + 1$ и переход к пункту 11, иначе переход к пункту 13.

13. На третьем этапе итерации t происходит общее испарение феромона на всех ребрах двудольного графа $H_{nm}(t)$ в соответствии с формулой $f_{ij} = f_{ij}(1 - \rho)$, где ρ – коэффициент обновления.

14. Находится агент с лучшим решением $F(t)$, полученным после выполнения t итераций, которое запоминается.

15. Если $t < NT$, то $t = t + 1$ и переход к пункту 6, иначе переход к пункту 16.

16. Конец работы алгоритма.

Рассмотрим теперь конструктивный алгоритм построения муравьем двудольного графа-решения $H^l(t)$ на полном двудольном графе $H_{nm}(t-1)$.

Алгоритм муравья.

1. $s=0$ (s – индекс списка вершин $W(s)$, указывающий на число удаленных вершин из исходного списка W).

2. Формируется исходный список вершин $W(s) = \{w_j / j=1, 2, \dots, m\}$, включающий все вершины множества W , то есть $W(s) = W = \{w_j / j=1, 2, \dots, m\}$.

3. В списке $W(s)$ выбирается вершина w_j , которая закреплена за муравьем a_l , и удаляется из списка. Переход к пункту 5.

4. Случайным образом в списке $W(s)$ выбирается вершина w_j , которая удаляется из списка.

5. $s = s + 1, w_j \in W$.

6. Для выбранной вершины w_j определяется набор ребер U_j , связывающих w_j со всеми вершинами e_i множества $E, |U_j|=n$.

7. Для каждого ребра $u_k \in U_j$, связывающего вершину e_i с вершиной w_j , определяется параметр f_{ij} – суммарный уровень феромона на этом ребре.

8. Для каждого ребра $u_k \in U_j$ по формуле $P_k = f_{ij} / \sum_i (f_{ij})$ рассчитывается вероятность включения ребра $u_k \in U_j$ в формируемый граф-решение H^l .

9. Агент с вероятностью P_k выбирает одно из ребер, которое включается в формируемый граф-решение $H^l(t)$.

10. Если $s < m$, то переход к пункту 4, иначе переход к пункту 11.

11. Двудольный граф-решение $H^l(t)$ полностью сформирован. Конец работы алгоритма муравья.

Разработка программы

Алгоритм решения задачи распределения работы программ на многопроцессорных системах был запрограммирован на языке *c#* на платформе

Windows. При этом все исследования проводились на компьютере типа Intel® Core™ i5 CPU 3.33 GHz и ОЗУ размером 4 Гб.

Язык C# был выбран по нескольким причинам. Во-первых, он позволяет быстро наладить разработку на начальных этапах, что дает возможность в короткие сроки спроектировать программный продукт, который потом можно будет отлаживать с помощью средств Visual Studio.

Огромное количество библиотек с .NET идет в базе, плюс к ним множество свободно доступных библиотек, что покрывает практически все перво-степенные задачи разработки под Windows. Наличие большого количества стандартных типов почти избавляет от библиотек, где базовые типы перепределены. В силу того, что библиотеки C# сравнительно молодые, интерфейсы библиотек, как правило, лучше вписываются в те или иные шаблоны проектирования.

В Visual Studio есть возможность подключения системы контроля версий и работы с ней. Эта особенность сильно помогает в процессе коллективной разработки.

Также данный язык изначально инкапсулирован, что позволяет создавать иерархические схемы наследования и делает возможным использование компонентно-ориентированного подхода.

Для проектирования программы был выбран компонентно-ориентированный подход. Это парадигма программирования, опирающаяся на понятие компонента – независимого модуля программного кода и предназначенная для повторного использования и развертывания, реализующегося в виде множества языковых конструкций.

Кроме того, в платформе .NET реализован компонентно-ориентированный подход, обеспечивающий создание и повторное использование компонентов для любого языка программирования, поддерживаемого платформой.

BaseDisplayer – класс, содержащий в себе виртуальные функции, предназначенные для отрисовки различных статистических элементов на формах (графиках и т.д.). От него наследуются классы DrawBestSolution, ProblemDisplayer, RAlgorithmDisplayer. DrawBestSolutions предназначен для отрисовки графика нахождения лучших решений по итерациям; ProblemDisplayer – для отображения текущей решаемой задачи на форме, предоставляющей возможность загрузки решения из файла; RAlgorithmDisplayer – для отображения информации о текущих настройках алгоритма.

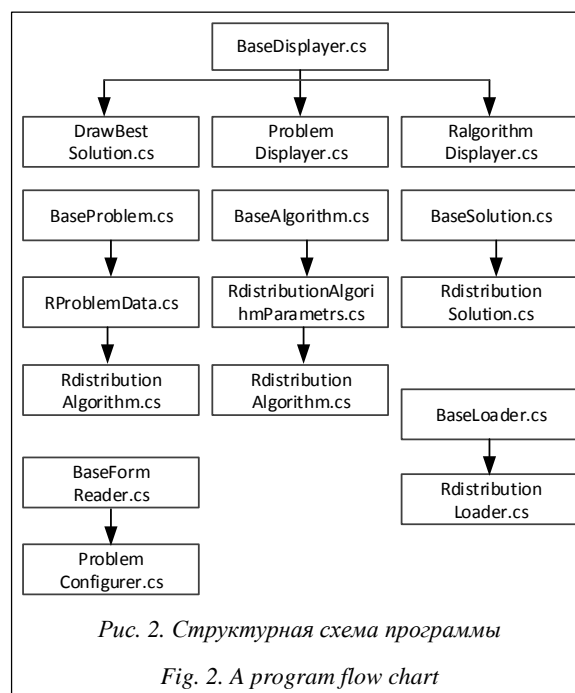
Программа содержит набор классов, предназначенных для обеспечения работы алгоритма распределения ресурсов.

Класс BaseAlgorithm. Самый верхний в иерархии наследования, общий для любого алгоритма, решающего задачу методом моделирования колонии муравьев. Он абстрактен и содержит определенные виртуальные функции GetProblem и

ResetAllParameters, а также ряд перечислителей с информацией о способе работы алгоритма.

Класс RDistributionAlgorithmParameters содержит основные параметры работы алгоритма: количество итераций, количество муравьев, параметры отложения феромона (начальное, отложение феромона муравьем, параметр затухания, способ отложения феромона). Феромон могут откладывать муравьи, нашедшие лучшее решение, абсолютно все муравьи, муравьи, нашедшие лучшее решение и решения, находящиеся в определенном оценочном диапазоне относительно лучшего.

На рисунке 2 приведена структурная схема программы, описывающая компонентное устройство программы.



Класс RDistributionAlgorithm определяет перечисленные функции самого муравьиного алгоритма и представляет основную программную логику его работы.

Класс BaseProblem содержит некоторые базовые данные о проблеме, решение которой происходит на момент работы алгоритма. Он верхний в иерархии наследования.

Класс RproblemData содержит все необходимые исходные данные для работы алгоритма.

Класс RDistributionAlgorithm наследуется от RproblemData. Это сделано для более упрощенного доступа ко всем исходным данным. Класс содержит реализацию всех механизмов, необходимых для работы алгоритма моделирования поведения колонии муравьев.

Класс BaseFormReader. Базовый класс, предназначенный для считывания информации об исходных данных с формы. Содержит только виртуальные функции.

Класс ProblemConfigurer. Конфигуратор исходных данных. Наследуется от BaseFormReader. Также имеет доступ к RdistributionProblem. Преобразует исходные данные задачи в формат, использующийся внутри программы.

Класс BaseLoader предназначен для загрузки файлов. Содержит только виртуальные функции.

Класс RdistributionLoader. Предназначен для загрузки данных с привязкой к задаче распределения программ в многопроцессорных системах. Реализует возможность подгрузки файлов из формата .xml или сериализованного бинарного файла.

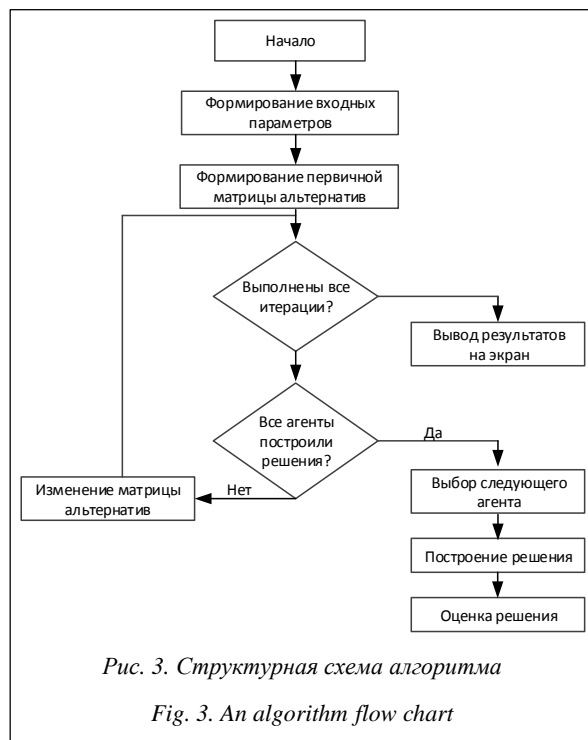
Класс BaseSolution. Представляет базовое построенное решение задачи внутри программы. Содержит только виртуальные функции.

Класс RdistributionSolution. Содержит информацию о полученном решении с привязкой к задаче распределения программ в многопроцессорных системах. Наследуется от BaseSolution.

Как видно из описания, программа содержит несколько полностью виртуальных классов, что позволяет использовать данный шаблон для последующих разработок.

Программная реализация алгоритма

Рассмотрим реализацию алгоритма моделирования поведения колонии муравьев (рис. 3).



Вначале производится формирование исходных данных. На этом шаге или загружается информация из файла, или специальными генераторами создается множество работ с заданным временем выполнения. Также задаются количество итераций работы алгоритма и количество агентов (муравьев).

Задаются параметры, обеспечивающие не-прямой обмен (стигмержи), такие как начальное количество феромона и количество феромона, которое способен отложить каждый отдельный агент системы. Существует ряд параметров, определяющих некоторые системы поведения муравьев. Каждый агент может двигаться, начиная с первой вершины, начиная со своей вершины, также он может выбирать вершины, по которым осуществляется движение в случайном порядке, руководствуясь генератором псевдослучайных чисел.

Затем осуществляется генерация первичной матрицы альтернатив. Матрица содержит значение вероятности, с которой муравей может выбрать какую-то конкретную вершину для своего движения. На первой итерации каждый элемент множества альтернатив приравняется к начальному количеству феромона.

Если не все итерации алгоритма выполнены и не все агенты осуществили процедуру построения решения, выбирается определенный агент, который реализует его построение. Далее оценивается решение. Оценка происходит по выбору максимально загруженного процессора.

Если все агенты на текущей итерации выполнили построение решений, производится изменение множества альтернатив. При этом каждое решение сравнивается с найденным лучшим решением на данной итерации. Чем хуже решение, тем меньшее количество феромона откладывает агент, который данное решение получил. Откладывание феромона в матрицу альтернатив осуществляется по формуле $Pd_i = \frac{Pd_i * Sb_i}{Sc_i}$, $i = \{0, 1, \dots, n\}$, где

Pd – количество откладываемого феромона муравьем на i -й итерации; Sb – оценка лучшего найденного решения; Sc – оценка текущего найденного решения.

Проведение экспериментальных исследований

При проведении экспериментальных исследований необходимо было определить две характеристики: эффективность полученного алгоритма и качество разработанных на основе муравьиной колонии механизмов, которые были применены для решения однородной распределительной задачи.

Для этого была применена процедура формирования контрольных тестовых примеров с уже известными результатами (оптимумом).

Первая исследуемая характеристика – влияние (на полученный результат) управляющих параметров, таких как размер популяции муравьев, количество итераций, параметров, управляющих отложением и испарением феромона.

Для получения достоверных выводов была проведена серия тестов-экспериментов. Временная зависимость разработанного алгоритма определяется

временем жизни колонии t (количество генераций), числом исполнителей n и числом работ m и определяется как $O(t \cdot n^2 \cdot m)$. Тестирование показало, что в 97 % случаев сформированное пространство решений содержит глобальное оптимальное решение.

Тестирование, определяющее сходимость алгоритма, выполнялось следующим образом. Для каждого теста запоминался номер итерации, после которой улучшения оценки не происходило. Проводилась серия из 50 тестов, в которой находились минимальный и максимальный номера итерации. Также выполнялся расчет среднего значения количества итераций, улучшения оценки после которых не происходило. Фактически в каждой серии тестов определялось лучшее решение, которое являлось оптимальным. Тестовые эксперименты показали, что сходжение алгоритма происходит на 120-й генерации при объеме популяции $M = 90$.

Сравнение значений критерия, полученных муравьиным алгоритмом на бенчмарках, у которых оптимум уже известен, показало, что у 80 % тестов полученный результат был оптимальным, у 15 % тестов результаты были на 3 % хуже оптимального, а у 5 % тестов результаты были хуже не более, чем на 2 %. На основании проведенных экспериментальных исследований можно сделать вывод, что разработанный алгоритм позволяет получить результаты на 2–3 % лучше тех, что получены имеющимися алгоритмами [2–10].

В процессе разработки в программу была добавлена возможность построения гистограмм для наилучшего определения итерации, на которой происходит сходимость алгоритма и нахождение лучшего решения. Пример гистограммы для некоторых вариантов решений приведен на рисунке 4.

Для сравнения эффективности были выбраны генетический алгоритм, *селективно-перестановочный алгоритм* (СПА) [4, 6–10], *алгоритм различных наборов работ и машин* (DJMS) [22], *алгоритм моделирования поведения колонии муравьев* (ACO).

Эти алгоритмы были выбраны в силу перспективности и хороших результатов, показанных ими при проведении экспериментов [4, 6–10].

Сравнивался алгоритм адаптивного поведения муравьиной колонии с различными алгоритмами решения однородной распределительной задачи. В качестве оптимальных настроек для алгоритма моделирования поведения колонии муравьев использовались следующие параметры: $N_c = 50$ – размерность колонии, $N_i = 100$ – количество итераций для каждого эксперимента. Работа алгоритма была смоделирована без учета элитных особей. Подобные параметры также использовались для алгоритма селективных перестановок.

Каждый проведенный эксперимент содержал три параметра: m – количество работников, n – количество работ, U – интервал работ, использующийся для генерации списка работ на каждой ите-

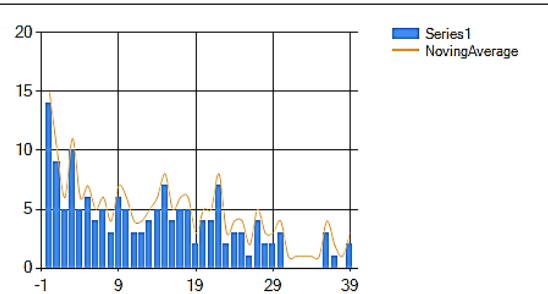


Рис. 4. Пример гистограммы (количество работ – 50, количество узлов – 3)

Fig. 4. An example of a histogram (a number of operations – 50, a number of nodes – 3)

рации. Для каждого набора работ использовались 100 задач. Суммарно для сравнения методов ACO и DJMS было проанализировано 1 900 решений.

Для оценки решений использовалась нижняя граница (Lower bound):

$$L = \max \left\{ \left[\frac{1}{m} * \sum_{j=1, \dots, n} p_j \right], p_1, p_m + p_{m+1} \right\}.$$

Сравним эффективность метода моделирования поведения адаптивной колонии с эффективностью методов DJMS и LPT (табл. 1). Результаты экспериментов метода DJMS взяты из [22]. Для каждого метода в каждой строчке таблицы показано количество оптимальных решений из проведенных 100 экспериментов.

Таблица 1

Сравнение эффективности методов

Table 1

Method effectiveness comparison

n	m	U	LPT	DJMS	ACO
6	3	[1, 20]	62	63	99
9	3	[1, 20]	35	79	98
15	3	[1, 20]	56	99	99
6	3	[20, 50]	45	45	99
9	3	[20, 50]	11	21	100
15	3	[20, 50]	26	48	99
8	4	[1, 20]	66	68	100
12	4	[1, 20]	34	76	98
20	4	[1, 20]	56	97	99
8	4	[20, 50]	40	40	97
12	4	[20, 50]	10	14	98
20	4	[20, 50]	24	45	100
10	5	[1, 20]	53	53	99
12	5	[1, 20]	48	79	100
25	5	[1, 20]	23	100	98
10	5	[20, 50]	47	22	97
12	5	[20, 50]	27	11	98
25	5	[20, 50]	9	31	100

Из таблицы 1 видно, что алгоритм DJMS опережает стандартный LPT. Стоит отметить, что этот

алгоритм использует связку концепций LPT и MF. Очевидно, что такая связка дает более хороший результат. В свою очередь, алгоритм АСО опережает DJMS, что обусловлено использованием вероятностного подхода и выцветанием отдельных альтернативных выборов решений.

Сравним алгоритмы АСО и СПА.

Для исследования эффективности этих алгоритмов проведены вычислительные эксперименты при разных значениях параметров задачи. В задаче приводятся такие же параметры, как и в предыдущем опыте, за исключением диапазона работ. В качестве параметра, характеризующего ресурсно-точные свойства, выбран $P_{опт}$ – доля оптимальных решений.

Таблица 2 позволяет сравнить алгоритмы СПА и АСО. В данном случае применялся алгоритм селективных перестановок, использующий одинарные перестановки. Данные для СПА взяты из [23].

Таблица 2

Сравнение СПА и АСО

Table 2

Comparison of selective-permutation algorithm and ACO

n	m	U	СПА	АСО
5	33	[35,65]	95	99
5	33	[15,85]	80	98
5	63	[35,65]	100	99
5	63	[15,85]	100	100
6	48	[25,75]	72	98
7	33	[35,65]	79	99
7	33	[15,85]	57	97
7	63	[35,65]	42	98
7	63	[15,85]	46	98

Алгоритм муравьиной колонии показал эффективность 98,6 %, в то время как эффективность алгоритма ListScheduling составила 21,9 %, ThresholdHeuristic – 79,1 % и Scattersearch – 97,7 % (результаты были взяты из [24]).

Стоит отметить, что алгоритм моделирования поведения колонии муравьев показывает хорошие результаты на абсолютно различных диапазонах мощностей работ, что связано с имитацией откладывания феромона во время работы алгоритма.

По сравнению с алгоритмом DJMS метод показывает гораздо более хороший результат благодаря использованию более мощной эвристики.

Методы СПА и АСО показывают приблизительно равные результаты с преимуществом метода АСО около 2,5 %. Также стоит отметить, что метод АСО показывает более хороший результат на большом диапазоне работ. В целом данный метод имеет преимущество.

Заключение

В работе предложена новая парадигма комбинаторной оптимизации, которая базируется на моде-

лировании адаптивного поведения муравьиной колонии и представляет графическое решение однородной распределительной задачи в виде двудольного графа. Также для решения однородных распределительных задач предложены новые механизмы. Муравьем на графе поиска решений строится двудольный граф в отличие от канонической парадигмы муравьиного алгоритма. В оптимизационных задачах, допускающих представление решения в виде двудольных графов, этот способ поиска рациональных решений является наиболее эффективным.

В частности, путем незначительной модификации конструктивного алгоритма муравья разработанный алгоритм, решающий однородные распределительные задачи, может также решать задачу о назначениях.

Подбор наиболее адекватных управляющих параметров может стать источником усовершенствования данного алгоритма. Решением другой проблемы, которая может улучшить критерий качества, является установление очередности рассматриваемых работ. Улучшение качества можно получить с помощью интеграции различных методов роевого интеллекта.

Работа выполнена при финансовой поддержке РФФИ, грант № 17-07-00997.

Литература

1. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы. М.: Изд-во МГУ, 2011. 188 с.
2. Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети. СПб: Энергоатомиздат, 1987. 329 с.
3. Романовский И.В. Алгоритмы решения экстремальных задач. М.: Наука, 1977. 156 с.
4. Нейдорф Р.А., Жикюлин А.А. Быстродействующая модификация алгоритма оптимизации решения // AIS-IT'14: тр. Конгресса по интелект. сист. и информ. технол. В 4-х т. М.: Физматлит, 2010. Т. 1. С. 15–22.
5. Кобак В.Г., Будиловский Д.М. Сравнительный анализ приближенных алгоритмов решения минимаксной задачи для однородных приборов // Вестн. Донск. гос. техн. ун-та. 2006. № 4. С. 327–334.
6. Нейдорф Р.А., Жикюлин А.А. Исследование вариантов модификации приближенных алгоритмов решения однородных распределительных задач, повышающих их эффективность // Инновация, экология и ресурсосберегающие технологии (ИнЭРТ-2012): тр. X Междунар. науч.-техн. форума. Ростов н/Д: Изд-во ДГТУ, 2012. С. 370–375.
7. Нейдорф Р.А., Жикюлин А.А. Селективно-перестановочный метод приближенного решения однородной распределительной задачи. Комбинаторные перестановки // Изв. ЮФУ. Сер. Технич. науки. 2013. № 7. С. 167–172.
8. Кобак В.Г. Эффективные методы решения однородных распределительных задач на основе минимаксного критерия. Ростов н/Д: Изд-во ДГТУ, 2013. 99 с.
9. Титов Д.В. Методы повышения эффективности алгоритмов решения распределительных минимаксных задач в однородных системах: дисс. ... канд. технич. наук. Ростов н/Д: Изд-во ДГТУ, 2010. 179 с.
10. Жикюлин А.А. Методы высокоэффективной ресурсной модификации алгоритма Романовского для решения однородных распределительных задач. дисс. ... канд. технич. наук. Ростов н/Д: Изд-во ДГТУ, 2014. 156 с.

11. Davidović T., Šelmić M., Teodorović D., Ramljak D. Bee colony optimization for scheduling independent tasks to identical processors. *Jour. of Heuristics* 2012, vol. 18, iss. 4, pp. 549–569.
12. Demirel T., Özkır V., Demirel N.Ç., Taşdelen B. A Genetic Algorithm approach for minimizing total tardiness in parallel machine scheduling problems. *Proc. World Congress on Engineering*, 2011, vol. II, pp. 267–281.
13. Raghavendral B.V., Murthy A.N. Workload balancing in identical parallel machine scheduling while planning in flexible manufacturing system using genetic algorithm. *ARNP Jour. of Engineering and Applied Sciences*, 2011, vol. 6, pp. 113–126.
14. Aslam M.U., Nasr M.M., Al-Harkan I.M. Scheduling of jobs on identical parallel machines to minimize total flow time subject to optimal makespan. *Applied Mechanics and Materials*, 2014, vol. 529, pp. 390–395.
15. Akylol D.E. Identical parallel machine scheduling with dynamical networks using time-varying penalty parameters. *Multiprocessor Scheduling, Theory and Applications*, I-Tech Education and Publishing, 2007, pp. 293–315.
16. Dorigo M. and Stützle T. *Ant colony optimization*. MIT Press, Cambridge, MA, 2004, 234 p.
17. Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования: монография. Таганрог: Изд-во ЮФУ, 2013. 199 с.
18. Raidl G.R. *A unified view on hybrid metaheuristics*. LNCS. Springer-Verlag, 2006, pp. 1–12.
19. Лебедев Б.К., Лебедев О.Б. Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // *Изв. ЮФУ*. 2012. № 7. С. 27–35.
20. Лебедев В.Б., Лебедев О.Б. Роевой интеллект на основе интеграции моделей адаптивного поведения муравьиной и пчелиной колоний // *Изв. ЮФУ*. 2013. № 7. С. 41–47.
21. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Разбиение на основе моделирования адаптивного поведения биологических систем // *Нейрокомпьютеры: разработка, применение*. 2010. № 2. С. 28–34.
22. Лебедев Б.К., Лебедев О.Б., Нацкевич А.Н. Решение однородной распределительной задачи методом моделирования адаптивного поведения колонии муравьев. Свид. о гос. регистр. прогр. для ЭВМ № 2017611229 от 27.01.2017.
23. A new algorithm to minimize makespan on identical parallel machines. URL: <http://www.pomsmeetings.org/confpapers/025/025-0543.pdf> (дата обращения: 09.04.2017).
24. Iori M., Martello S. Scatter search algorithms for identical parallel machine scheduling problems. URL: http://www.springer.com/cda/content/document/cda_downloaddocument/9783540789840-c1.pdf?SGWID=0-0-45-552307-p173814816 (дата обращения: 09.04.2017).

Software & Systems

DOI: 10.15827/0236-235X.118.217-226

Received 10.04.17

2017, vol. 30, no. 2, pp. 217–226

A HOMOGENEOUS DISTRIBUTION PROBLEM BASED ON ANT COLONY ADAPTIVE BEHAVIOOR MODELS

B.K. Lebedev¹, *Dr.Sc. (Engineering), Professor, lebedev.b.k@gmail.com*

O.B. Lebedev¹, *Ph.D. (Engineering), Associate Professor, lebedev.ob@mail.ru*

E.M. Lebedeva¹, *Postgraduate Student, lebedeva.el.m@mail.ru*

¹ *Institute of Computer Technology and Information Security, Southern Federal University, Engels St. 1, Taganrog, 347928, Russian Federation*

Abstract. The paper proposes a solution of a homogeneous distribution problem. It gives the problem statement, describes the main groups of algorithms to solve it (approximate and exact) and their advantages and disadvantages. The paper proposes a new paradigm of combinatorial optimization, which is based on modeling the adaptive behavior of an ant colony.

The solution of a homogeneous distribution problem is its graphical representation as a bipartite graph. New decision mechanisms were proposed to solve these problems. The basis of metaheuristics of an ant colony algorithm is a combination of two techniques. The first basic technique is to perform the search for the best solution using an ant colony adaptive behavior. An ant builds a specific solution using a built-in procedure, which is based on a constructive algorithm. A bipartite graph that is built on the solution search graph is the main difference of the proposed ant algorithm from the existing canonical paradigm.

When finding optimal solutions for optimization problems, which allow presenting solutions in the form of bipartite graphs, this approach will be fairly effective.

The conducted researches showed that the ant algorithm gives more qualitative solutions in comparison with the known algorithms. Comparing known and developed algorithms, we can say that the results improved by 3–4 %.

Keywords: assignment problem, bipartite graph, optimization, swarm intelligence, ant colony, adaptive behavior, homogeneous distribution problem.

Acknowledgements. The work has been financially supported by the Russian Foundation for Basic Research, grant no. 17-07-00997.

References

1. Lazarev A.A., Gafarov E.R. *Teoriya raspisany. Zadachi i algoritmy* [The Scheduling Theory. Tasks and Algorithms]. Moscow, Lomonosov MSU Publ., 2011.
2. Mayorov S.A., Novikov G.I. *Vychislitelnye komplekxy, sistemy i seti* [Computer Complexes, Systems and Networks]. St. Petersburg, Energoatomizdat Publ., 1987.

3. Romanovsky I.V. *Algoritmy resheniya ekstremal'nykh zadach* [Solution Algorithms for Extremum Problems]. Moscow, Nauka Publ., 1977.
4. Neydorf R.A., Zhikulin A.A. A Quick-operating modification of the Solution optimization algorithm. *AIS-IT'14: tr. Kongr. po intellekt. sist. i inform. tekhnol.* [Proc. Congr. on Intelligent Systems and IT]. Moscow, Fizmatlit Publ., 2010, vol. 1, pp. 15–22 (in Russ.).
5. Kobak V.G., Budilovsky D.M. A comparative analysis of approximate algorithms to solve a minimax problem for homogeneous instruments. *Vestn. DGTU* [Vestnik of DSTU]. 2006, no. 4, pp. 327–334 (in Russ.).
6. Neydorf R.A., Zhikulin A.A. A research on variations of approximate algorithms for solving a homogeneous distribution problems, which increase their effectiveness. *Innovatsiya, ekologiya i resursoberegayushchie tekhnologii (InERT-2012): tr. X Mezhdunar. nauch.-tekhn. foruma* [Proc. 10th Int. Science and Technical Forum on Innovation, Ecology and Resource-Saving Technologies (InERT-2012)]. Rostov-on-Don, DGTU Publ., 2012, pp. 370–375 (in Russ.).
7. Neydorf R.A., Zhikulin A.A. Selective-permutation method for finding an approximate solution of the open shop scheduling problem. Combinational permutations. *Izv. YuFU. Ser. Tekhnich. nauki* [Izvestiya SFedU. Engineering Sciences]. 2013, no. 7, pp. 167–172 (in Russ.).
8. Kobak V.G. *Effektivnye metody resheniya odnorodnykh raspredelitel'nykh zadach na osnove minimaksnogo kriteriya* [Efficient Methods for Solving Homogeneous Distribution Problems Based on a Minimax Criterion]. Rostov-on-Don, DGTU Publ., 2013, 99 p.
9. Titov D.V. *Metody povysheniya effektivnosti algoritmov resheniya raspredelitel'nykh minimaksnyykh zadach v odnorodnykh sistemakh* [Improving Algorithm Practices of Distribution Minimax Problems in Homogeneous Systems]. PhD thesis. Rostov-on-Don, DGTU Publ., 2010.
10. Zhikulin A.A. *Metody vysokoeffektivnoy resursnoy modifikatsii algoritma Romanovskogo dlya resheniya odnorodnykh raspredelitel'nykh zadach* [The Methods of Highly Efficient Resource Modification of Romanovsky Algorithm to Solve Homogeneous Distribution Problems]. PhD thesis. Rostov-on-Don, DGTU Publ., 2014.
11. Davidović T., Šelmić M., Teodorović D., Ramljak D. Bee colony optimization for scheduling independent tasks to identical processors. *Jour. of Heuristics*. 2012, vol. 18, iss. 4, pp. 549–569.
12. Demirel T., Özkır V., Demirel N.Ç., Taşdelen B. A Genetic Algorithm approach for minimizing total tardiness in parallel machine scheduling problems. *Proc. World Congr. on Engineering*. 2011, vol. 2.
13. Raghavendra B.V., Murthy A.N. Workload balancing in identical parallel machine scheduling while planning in flexible manufacturing system using genetic algorithm. *ARNP Jour. of Engineering and Applied Sciences*. 2011, vol. 6.
14. Aslam M.U., Nasr M.M., Al-Harkan I.M. Scheduling of jobs on identical parallel machines to minimize total flow time subject to optimal makespan. *Applied Mechanics and Materials*. 2014, vol. 529, pp. 390–395.
15. Akyol D.E. Identical parallel machine scheduling with dynamical networks using time-varying penalty parameters. *Multiprocessor Scheduling, Theory and Applications*. I-Tech Education and Publ., 2007, pp. 293–315.
16. Dorigo M., Stützle T. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
17. Lebedev O.B. *Modeli adaptivnogo povedeniya muravinoy kolonii v zadachakh proektirovaniya* [Ant Colony Adaptive Behavior Models in Design Problems]. Monograph. Taganrog, SFedU Publ., 2013, 199 p.
18. Raidl G.R. A unified view on hybrid metaheuristics. *LNCS*. Springer-Verlag Publ., 2006, pp. 1–12.
19. Lebedev B.K., Lebedev O.B. Modelling of an ant colony adaptive behaviour by search of the decisions interpreted by trees. *Izv. YuFU. Ser. Tekhnich. nauki* [Izvestiya SFedU. Engineering Sciences]. 2012, no. 7, pp. 27–35 (in Russ.).
20. Lebedev V.B., Lebedev O.B. Swarm intelligence on the basis of the adaptive behaviour models integration of the ant and bee colonies. *Izv. YuFU. Ser. Tekhnich. nauki* [Izvestiya SFedU. Engineering Sciences]. 2013, no. 7, pp. 41–47 (in Russ.).
21. Kureychik V.M., Lebedev B.K., Lebedev O.B. Partitioning on the basis of modeling of adaptive behaviour of biological systems. *Neyrokomp'yutery: razrabotka, primeneniye* [Neurocomputers: Design and Applications]. 2010, no. 2, pp. 28–34 (in Russ.).
22. Lebedev B.K., Lebedev O.B., Natskevich A.N. *Reshenie odnorodnoy raspredelitel'noy zadachi metodom modelirovaniya adaptivnogo povedeniya kolonii muravev* [Solving a Homogeneous Distribution Problem by the Method of Modeling Ant Colony Adaptive Behavior]. State Registration Certificate of a computer program no. 2017611229. 2017.
23. *A New Algorithm to Minimize Makespan on Identical Parallel Machines*. Available at: <http://www.pomsmeetings.org/confpapers/025/025-0543.pdf> (accessed April 9, 2017).
24. Iori M., Martello S. *Scatter Search Algorithms for Identical Parallel Machine Scheduling Problems*. Available at: http://www.springer.com/cda/content/document/cda_downloaddocument/9783540789840-c1.pdf?SGWID=0-0-45-552307-p173814816 (accessed April 9, 2017).

Примеры библиографического описания статьи

1. Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. Однородная распределительная задача на основе моделей адаптивного поведения муравьиной колонии // Программные продукты и системы. 2017. Т. 30. № 2. С. 217–226; DOI: 10.15827/0236-235X.118.217-226.
2. Lebedev B.K., Lebedev O.B., Lebedeva E.M. A homogeneous distribution problem based on ant colony adaptive behavior models. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 217–226 (in Russ.); DOI: 10.15827/0236-235X.118.217-226.

УДК 004.042

DOI: 10.15827/0236-235X.118.227-234

Дата подачи статьи: 19.12.16

2017. Т. 30. № 2. С. 227–234

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБРАБОТКИ ДАННЫХ В UNIX-ПОДОБНЫХ СИСТЕМАХ

Е.В. Пальчевский, аспирант, teelxp@inbox.ru;

*А.Р. Халиков, к.ф.-м.н., доцент, khalikov.albert.r@gmail.com
(Уфимский государственный авиационный технический университет,
ул. Карла Маркса, 12, г. Уфа, 450008, Россия)*

Данная статья посвящена распределенно-модульной обработке информации в автоматизированном режиме. Эта разработка позволяет принимать входящие и исходящие данные на физический сервер в объеме до 2,2 GB/s, при этом распределяя потоковую информацию (весь входящий сетевой трафик на сервер) по физическим и логическим ядрам.

Показана нагрузочная зависимость физических ресурсов от входящей информации. Обоснована целесообразность применения разработанного аппаратно-программного комплекса SDP (Speed data processing), а также представлены структура и принципиальная схема работы. На первом этапе создания комплекса осуществляется разработка алгоритма, на втором – его техническая реализация. Приведен фрагмент исходного кода, отвечающего за уведомление на e-mail как о нагрузке на центральный процессор, так и об основных запускаемых процессах. Описан основной функционал со следующими данными: название функции, цель функции, теоретическая нагрузка, лимит передачи данных (в MB/s) и результат выполнения. На третьем этапе проводится тестирование комплекса SDP, представлены среднесуточные результаты за десять дней.

Созданный аппаратно-программный комплекс позволяет эффективно обрабатывать входящую и исходящую информацию в автоматическом режиме для увеличения пропускной способности при приеме и отправке данных в СУБД MySQL, в том числе при DoS- и DDoS-атаках. Одной из частей комплекса является web-модуль для уведомления как с персонального компьютера, так и с мобильного телефона. В мониторинговой части web-модуля реализована возможность уведомления SMS-сообщениями о состоянии загруженности физического сервера.

Разработанный аппаратно-программный комплекс показал высокую стабильность при обработке больших объемов данных с минимальной нагрузкой на ЭВМ.

Ключевые слова: обработка информации, нагрузка на вычислительные ресурсы, организация данных, снижение нагрузки информационных потоков, потоки информации, SMS-уведомление, распределенная система, центр обработки данных.

Одним из активно развивающихся направлений является использование АСУ при передаче и обработке данных. Подобными вопросами зачастую занимаются дата-центры (*центры обработки данных* (ЦОД)) [1]. Такие центры работают в крупных городах во многих странах, и их число постоянно растет. ЦОД являются местом для хранения и обработки информации. В них располагаются различные ресурсы, такие как web-сайты, почтовые сервисы, облачные технологии и др. В настоящее время ЦОД нередко подвергаются атакам во внешней глобальной сети, которые могут исходить из любой точки планеты. Расположенное в ЦОД оборудование использует внешние сетевые каналы, что создает потенциальную угрозу информационной безопасности. При этом внешние интернет-ресурсы становятся недоступными для клиентов, например, пользователь не может попасть на свой почтовый сервис или не имеет доступа к нужной веб-странице и т.п.

На оборудовании, размещенном в ЦОД, как правило, установлены операционные системы различных семейств: Windows, Linux, UNIX, Amiga, DOS, Solaris, Illumos, Plan, Inferno [2]. Практически каждая система имеет определенные уязвимости, и нельзя быть застрахованным от DoS- и DDoS-атак, приводящих к нарушению доступности информации и потере пакетов. Правильное распределение информационных потоков по ядрам физического

сервера способствует увеличению пропускной способности при обработке данных в автоматизированном режиме и их сохранении. Распределенно-модульная обработка информации в автоматизированном режиме позволяет принимать входящие и исходящие данные на физический сервер до 2,2 GB/s, при этом распределяя потоковую информацию по физическим и логическим ядрам.

Главным требованием к современным технологиям является равномерное распределение выходных данных. Это означает, что упорядоченная входящая информация (например, потоковый сетевой трафик) должна преобразовываться автоматически. Подобная система автоматизации должна исключать энтропию преобразуемых данных [3].

Вопросами изменения автоматизированных информационных систем занимаются уже продолжительное время, и сегодня данным исследованиям посвящено огромное количество публикаций [3]. Одна из первых работ датируется 1980 годом. Ее авторами являются сотрудники корпорации IBM [4]. С тех пор автоматизации обрабатываемой информации придается огромное значение. В таких системах зачастую предлагаются устаревшие подходы к решению поставленной задачи, нагружающие вычислительные ресурсы [5]. Среднегодовая нагрузка на физические данные ЭВМ имеет тенденцию к ежегодному росту [6]. При огромных потоках информации увеличивается риск потери до-

ступности данных, находящихся на вычислительных ресурсах, которые имеют доступ во внешнюю глобальную сеть. Подобный эффект объясняется затруднениями в выявлении и классификации таких угроз, а также сложностью привязки их к конкретным процессам взаимодействия приложений в логируемых файлах [7].

В статье описана разработка *аппаратно-программного комплекса* (АПК) для обработки массивной и многопоточной информации (атака сетевым трафиком, парсинг, различные подключения от пользователей и т.п.) с последующим занесением данных в СУБД MySQL, почтовым и SMS-уведомлением, а также с выводом в web-интерфейс и распределением сетевой нагрузки по физическим и логическим ядрам сервера.

Преимущества и научная новизна

Основные преимущества предлагаемого решения:

- возможность распределения входящих информационных потоков (сетевой нагрузки) по ядрам физического сервера;
- увеличение пропускной способности СУБД MySQL;
- уведомление по SMS на мобильный телефон, а также на электронную почту;
- возможность удаленного управления АПК через web-интерфейс;
- увеличение сетевой пропускной способности (входящих и исходящих сетевых пакетов);
- использование цепей Маркова, впервые доработанных для распределения сетевой нагрузки.

Научная новизна заключается в распределении сетевой нагрузки по физическим ядрам сервера за счет доработки цепей Маркова, что показано на рисунках 1 и 2.

Таким образом, АПК не только позволяет снизить сетевую нагрузку на вычислительные ресурсы, но и способствует своевременному уведомлению (о состоянии ЭВМ) по SMS на мобильный телефон.

Аналогичные решения

Аппаратно-программные (автоматизированные) *системы обработки данных* (СОД) предназначены для создания информационных ресурсов (вычислительные кластеры). Обработка данных происходит на физическом (аппаратном) уровне. Аппаратно-программные СОД могут быть следующими.

1. Одномашинные СОД. Данное решение реализовано за счет единственной ЭВМ (физического сервера) с однопроцессорной структуризацией. Существенными недостатками разработанной системы являются ограничение производительности (до 10 млн операций в секунду), а также отсутствие

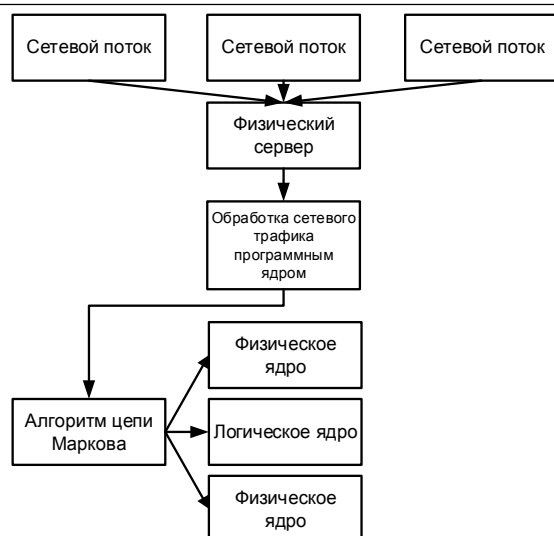


Рис. 1. Взаимодействие алгоритма цепи Маркова с информационными потоками

Fig. 1. Interaction of Markov chain algorithm with information flows

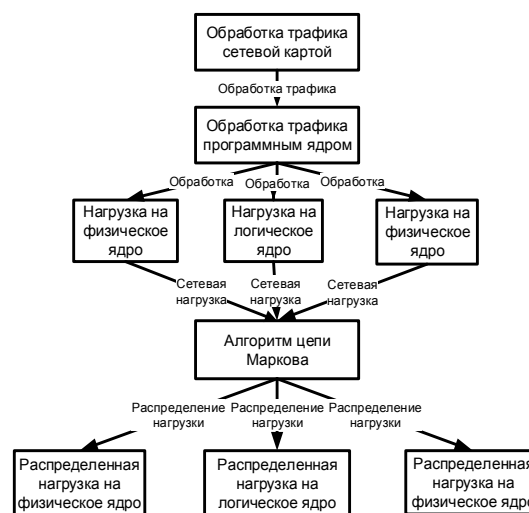


Рис. 2. Распределенная сетевая нагрузка за счет алгоритма цепи Маркова

Fig. 2. Distributed network load due to Markov chain algorithm

многопоточности приема входящего сетевого трафика [8]. В связи с этими недостатками возможен простой системы до 2–3 минут в сутки (по причине отказа в обслуживании оборудования). Структура одномашинной СОД представлена на рисунке 3.

2. Многомашинные СОД (вычислительные комплексы). Это решение предполагает связку нескольких физических серверов в единое целое. Данные вычислительные комплексы подразделяются на два вида: с прямой (рис. 4) и с косвенной (рис. 5) связями между ЭВМ. В СОД с косвенной связью взаимодействие происходит через общее запоминающее устройство (доступ с общим набо-

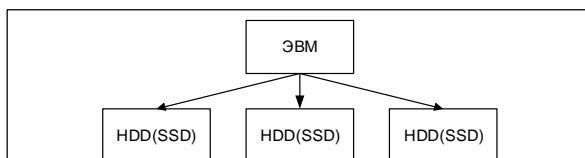


Рис. 3. Одномашинная СОД

Fig. 3. A single-machine data processing system

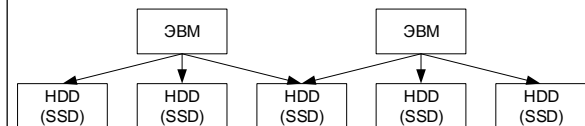


Рис. 4. Многомашинная СОД с косвенной связью

Fig. 4. A multiple-machine data processing system with indirect coupling

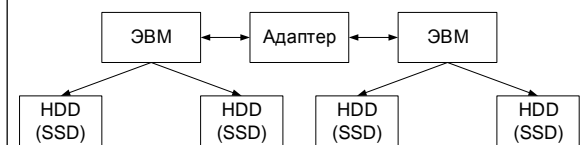


Рис. 5. Многомашинная СОД с прямой связью

Fig. 5. A multiple-machine data processing system with direct coupling

ром данных). В СОД с прямой связью процессы, отвечающие за работу с информацией, имеют доступ ко всему объему данных [9]. Данные СОД являются базовыми средствами обработки информации. В связи с этим в них включено универсальное и стандартизированное (общесистемное) ПО.

Существенным недостатком многомашинной СОД с косвенной связью является низкая пропускная способность принятия и обработки данных [10]. Минус многомашинной СОД с прямой связью: при выходе из строя одной из составляющих (физический сервер) замедляется скорость принятия и обработки данных.

3. СОД, настроенные на заданную область применения (вычислительные системы). Схема работы вычислительных систем показана на рисунке 6.

Все эти СОД основаны на взаимодействии между физическими серверами и вычислительными кластерами. Как правило, такие СОД включают в себя ПО и технические средства, ориентированные на решение определенных поставленных задач [11]. СОД (вычислительные комплексы) подразделяются на два вида:

- вычислительные системы на основе ЭВМ и вычислительных комплексов общего применения;
- вычислительные системы на основе специализированных ЭВМ и вычислительных комплексов.

Недостатком вышеприведенных видов СОД является возможность загрузки ресурсов центрального процессора, что может привести к отказу и задержке обработки данных [12].

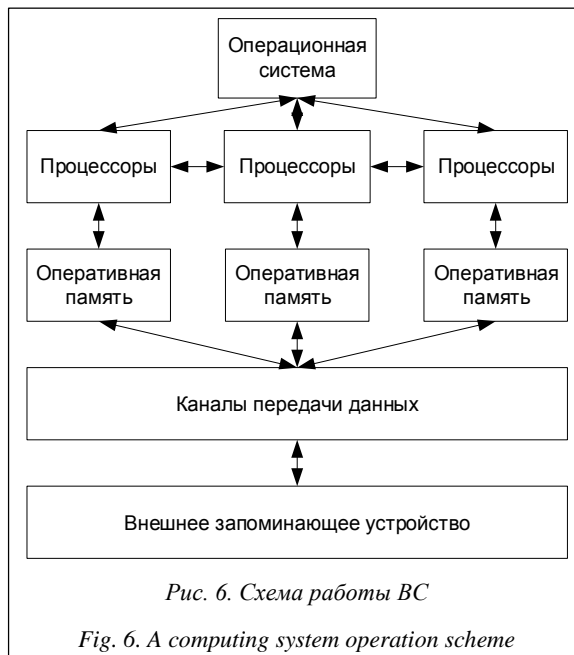


Рис. 6. Схема работы ВС

Fig. 6. A computing system operation scheme

4. Системы телеобработки (ТС). Данное решение предназначено для передачи данных по каналам связи: происходит взаимодействие пользователей с системой за счет терминалов связи (абонентских пунктов). Абонентские пункты подключаются через каналы связи к средствам обработки данных на ЭВМ [13–15]. Схема работы СОД ТС представлена на рисунке 7.

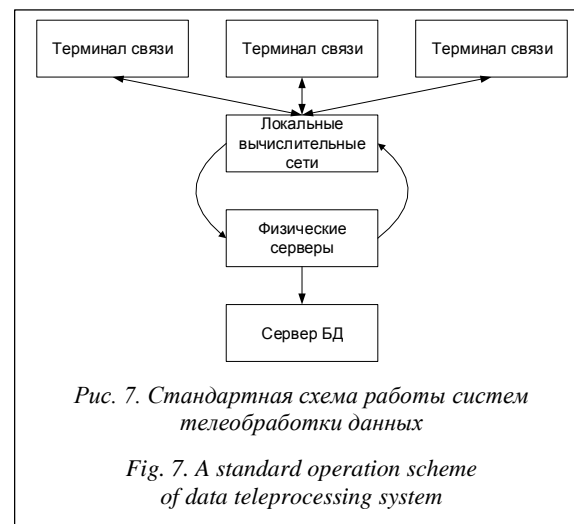


Рис. 7. Стандартная схема работы систем телеобработки данных

Fig. 7. A standard operation scheme of data teleprocessing system

Существенным недостатком подобного рода СОД является невозможность программного расширения пропускной способности канала для увеличения объема принимаемой и обрабатываемой информации.

Разработка и реализация автоматизированной обработки информации

Система SDP (Speed data processing) – разрабатываемый АПК, предназначенный для массивной и

Таблица 1

Функционал SDP и его нагрузка на вычислительные ресурсы ЭВМ

Table 1

SDP functionality and its load on computing resources

Название функции	Цель функции	Теоретическая нагрузка, %	Лимит передачи данных, МВ/с	Результат функций
Анализ данных	Анализ поступающей с различных датчиков информации	0,03	450	Прием обрабатываемых данных в режиме автоматизации
Проверка нагрузки	Сверка нагрузки с заданной нормой и оптимизация вычислительных ресурсов	0,10	690	Сверка нагрузки с последующей оптимизацией вычислительных ресурсов ЭВМ
Проверка внешнего сетевого интерфейса	Проверка сетевой активности на аномалии	0,05	1110	Проверка на аномалии сетевого трафика и их устранение
Общий анализ состояния ЭВМ	Проверка работоспособности физического сервера	0,02	1650	Проверка работоспособности и уведомление пользователя
Считывание логируемых файлов	Считывание информации	0,01	2200	Обработка информации в автоматизированном режиме и отправка в MySQL

бой очистку логируемых файлов после какой-либо DoS- и DDoS-атаки:

```
#!/bin/bash
filename=ddos.log
cat /dev/null > $filename; echo "Очистка созданного файла"
tail /var/log/messages > $filename
echo "В $filename была произведена запись об отчетно-сти прошедшей атаки"
```

Далее приведена часть исходного кода, созданная на языке программирования PHP и отвечающая за вызов функций в web-интерфейсе:

```
<?php
function microTimer_start()
{
    global $starttime;
    $mtime = microtime();
    $mtime = explode( ' ', $mtime );
    $mtime = $mtime[1] + $mtime[0];
    $starttime = $mtime;
}
function microTimer_stop()
{
    global $starttime;
    $mtime = microtime();
    $mtime = explode( ' ', $mtime );
    $mtime = $mtime[1] + $mtime[0];
    $endtime = $mtime;
    $totaltime = round( ($endtime - $starttime), 5 );
    return $totaltime;
}
function filters_input($check = 'all')
{
    $safehtml = new safehtml();
    $safehtml->protocolFiltering = "black";
    $safeinput = new safeinput();
    $safeinput->safeinput_check($check);
    unset($safehtml);
    unset($safeinput);
}
```

```
function filters_input_one($data, $type = "black", $nl2br =
= false, $LB_specchar = false)
{
    $safehtml = new safehtml();
    $safehtml->protocolFiltering = $type; // white или black
    $data = $safehtml->parse($data, $nl2br, $LB_specchar);
    unset($safehtml);
    return $data;
}
```

Фрагмент исходного кода, написанный на языке программирования высокого уровня C и отвечающий за распределение сетевой нагрузки по потокам:

```
bool execute_strict_cpu_affinity = false;
bool use_multiple_fanout_processes = true;
int main() {
    int fanout_group_id = getpid() & 0xffff;
    boost::thread speed_printer_thread( speed_printer );
    if (use_multiple_fanout_processes) {
        boost::thread_group packet_receiver_thread_group;
        unsigned int num_cpus = CPU;
        for (int cpu = 0; cpu < num_cpus; cpu++) {
            boost::thread::attributes thread_attrs;
            if (execute_strict_cpu_affinity) {
                cpu_set_t current_cpu_set;
                int cpu_to_bind = cpu % num_cpus;
                CPU_ZERO(&current_cpu_set);
                CPU_SET(cpu_to_bind, &current_cpu_set);
                Int set_affinity_result = pthread_attr_setaffinity_np(thread_attrs.native_handle(), sizeof(cpu_set_t), &current_cpu_set);
                if (set_affinity_result != 0) {
                    printf("It is impossible to parallelize loading on physical kernels\n");
                }
            }
            packet_receiver_thread_group.add_thread(
                new boost::thread(thread_attrs, boost::bind(start_
af_packet_capture, "eth6", fanout_group_id))
            );
        }
    }
}
```

Примеры SMS-уведомлений.

• В случае стабильной работы физического сервера (приходит ежечасное оповещение на мобильное устройство):

Дата: 10.08.2016;

Время: 12:00;

Размер потребляемого трафика на данный момент: 9 ТБ;

Текущая нагрузка: 5 %, в пределах нормы;

Состояние кластера стабильное.

• При каком-либо сбое ЭВМ, а также при наличии сетевой атаки:

Дата: 19.08.2016;

Время: 05:35;

Размер потребляемого трафика на данный момент: 34 ТБ;

Текущая нагрузка: 75 %, выше нормы!

Необходимо срочно принять меры!

Тестирование АПК

После подготовки алгоритма и написания кода АПК SDP необходимо произвести его тестирование для получения значений нагрузки на вычислительные ресурсы физического сервера в двух режимах – обычном и боевом. Обычный режим работы проявляется при упорядоченной (нормализованной) работе ЭВМ без воздействия внешних факторов. Под боевой работой подразумевается работоспособность ЭВМ с применением вредоносного сетевого трафика, направленного на нарушение удаленной работы серверного пространства.

Нагрузка на физические ресурсы ЭВМ при отсутствии каких-либо внешних факторов (обычный режим) представлена в таблицах 2 и 3.

В данных таблицах приняты следующие сокращения: А – количество обрабатываемой информации, МВ/с; В – количество потоков, шт./с; С – нагрузка при старте, %; D – нагрузка при рестарте, %; E – нагрузка при обработке информации, %; F – входящие сетевые пакеты, млн шт./с.

Средняя нагрузка из данных таблиц 2 и 3 показывает достаточно низкие результаты: центральный процессор – 0,77 %, SSD-накопитель – 0,53 %, оперативная память – 0,15 %.

Таблица 2

Потребление вычислительной мощности центрального процессора в течение десяти дней

Table 2

Consumption of processing capacity by CPU during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	50	100	150	200	250	350	450	580	760	900
B	5	10	15	20	25	30	35	40	45	50
C	0,10	0,19	0,24	0,35	0,46	0,59	0,64	0,78	0,99	1,15
D	0,11	0,20	0,25	0,36	0,47	0,60	0,65	0,79	1,00	1,16
E	0,14	0,28	0,42	0,56	0,70	0,84	0,98	1,12	1,26	1,40
F	0,22	0,44	0,66	0,88	1,10	1,32	1,54	1,76	1,98	2,20

Таблица 3

Нагрузка на твердотельный накопитель (SSD) в течение десяти дней

Table 3

Load on a solid-state drive (SSD) during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	50	100	150	200	250	350	450	580	760	900
B	5	10	15	20	25	30	35	40	45	50
C	0,08	0,13	0,18	0,24	0,35	0,46	0,52	0,69	0,78	0,85
D	0,09	0,14	0,19	0,25	0,36	0,47	0,53	0,70	0,79	0,86
E	0,11	0,22	0,33	0,44	0,55	0,66	0,77	0,88	0,99	1,10
F	0,12	0,24	0,36	0,48	0,60	0,72	0,84	0,96	1,08	1,20

Полученные данные (табл. 2, 3) позволяют констатировать, что нагрузка на вычислительные ресурсы ЭВМ достаточно низкая и предоставляет возможность широкомасштабной обработки данных с последующей стабилизацией серверной рабочей среды.

Нагрузка на ресурсы физического сервера при интенсивной атаке вредоносным сетевым трафиком (DoS и DDoS) показана в таблицах 4 и 5.

Таблица 4

Нагрузка на центральный процессор в течение десяти дней

Table 4

CPU usage during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	0,1	0,3	0,5	0,8	1,0	1,2	1,4	1,6	1,9	2,2
B	20	40	60	80	100	120	140	160	180	200
C	2,0	4,0	6,0	8,0	10,0	12,0	14,0	16,0	18,0	19,0
D	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
E	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
F	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0
G	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0
H	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0	13,0

Таблица 5

Нагрузка на SSD-накопитель в течение десяти дней

Table 5

Load on SSD-accumulator during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	0,1	0,3	0,5	0,8	1,0	1,2	1,4	1,6	1,9	2,2
B	20	40	60	80	100	120	140	160	180	200
C	2,0	4,0	6,0	8,0	10,0	12,0	14,0	16,0	18,0	19,0
D	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
E	0,55	0,6	0,7	0,9	1,18	1,45	1,67	1,88	2,03	2,06
F	0,56	0,7	0,8	1,0	1,19	1,46	1,68	1,89	2,04	2,07
G	0,8	1,6	2,4	3,2	4,0	4,8	5,6	6,4	7,2	8,0
H	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0

В данных таблицах приняты следующие обозначения: А – количество обрабатываемой информации, GB/s; В – количество потоков, шт./с; С – скорость атаки, GB/s; D – входящие сетевые пакеты, млн шт./с; Е – нагрузка при старте, %; F – нагрузка при рестарте, %; G – нагрузка при обработке информации, %; H – нагрузка при отправке данных в СУБД MySQL, %.

Средняя нагрузка (табл. 4, 5) остается достаточно низкой при многопоточной обработке информации большого объема: центральный процессор – 7,00 %, SSD-накопитель – 3,14 %, оперативная память – 0,75 %.

Результаты, приведенные в таблицах 4 и 5, свидетельствуют о высокой производительности АПК SDP. Таким образом, разработанный комплекс позволяет обрабатывать значительный объем информации и в режиме реального времени отправлять его в СУБД MySQL. Также появляется возможность запуска сложных вычислительных процессов без нарушения рабочей среды ЭВМ и при достаточно высокой производительности разработанного АПК.

Тестирование АПК SDP проводилось на следующем оборудовании: процессор 2xIntel Xeon 5670 (24 потока); оперативная память – 32 GB; винчестеры – RAID 10 (Intel S3710 SSDSC2BA012T401 800 GB каждый); суммарный сетевой канал – 5 GB/s; операционная система UBUNTU 14.04.

После апробации в течение одного месяца АПК SDP был окончательно инсталлирован на физический сервер в ЦОД (в г. Москва) на оборудование со следующими характеристиками: количество физических серверов – 30; процессор Intel Xeon 5660 (CPU – 60, физических ядер – 360, количество потоков – 720); оперативная память – 960 GB; винчестеры – RAID 10 (Intel S3710 SSDSC2BA012T401 800 GB каждый); внешний сетевой канал – 20 GB/s.

Заключение

В ходе проведенных исследований и разработок получены следующие результаты.

Разработан АПК по обработке потоковой информации в больших объемах. Это способствует совершенствованию пропускной способности между физическим сервером и реляционной БД MySQL.

Обоснована целесообразность применения АПК SDP на вычислительных ресурсах различной мощности. Предложена методика для упорядоченной и последовательной обработки информации на серверных вычислительных ресурсах.

Проведено множество тестирований, позволяющих определить эффективность работы АПК SDP. Средняя нагрузка на ресурсы центрального процессора варьируется от 0,77 % до 7,00 %. Данная тен-

денция предоставляет возможность запуска вычислительных процессов различного уровня, а также создает производительную рабочую среду, несмотря на большой поток обрабатываемой информации (до 2,2 GB/s). Загруженность твердотельного накопителя (SSD) колеблется в диапазоне от 0,53 до 3,14 %. Небольшая нагрузка на ячейки Solid-State Drive предоставляет возможность увеличения пропускной способности информационного канала передачи данных. Среднее потребление оперативной памяти составляет от 0,15 % до 0,75 %. Низкая нагрузка на вычислительные ресурсы позволяет выполнять скоростную обработку информации в автоматизированном режиме с последующим уведомлением (на выбор) системного администратора.

Таким образом, выявлена, проанализирована и исследована рабочая среда физического сервера с последующим переходом в кластер. Удобство в управлении и обработка информации на серверных вычислительных ресурсах позволили организовать стабильную работу ЭВМ.

Литература

1. Palchevsky E.V., Khalikov A.R. TCP/IP network STACK optimization under high load on UNIX-like systems. Proc. DSPTech'2015. USATU Publ., 2015, vol. 1, pp. 130–135.
2. Пальчевский Е.В., Халиков А.Р. Техника инструментирования кода и оптимизация кодовых строк при моделировании фазовых переходов на языке C++ // Труды Ин-та системного программирования РАН. 2015. Т. 27. № 6. С. 87–96.
3. Пальчевский Е.В., Халиков А.Р. Равномерное распределение нагрузки аппаратно-программного ядра в UNIX-системах // Труды Ин-та системного программирования РАН. 2016. Т. 28. № 1. С. 93–102.
4. IBM a management system for the information business. White Plains, NY, IBM Publ., 1980.
5. Olifer V.G., Olifer N.A. Computer networks. Principles, technologies, protocols. St. Petersburg, 2007, 960 p.
6. Колисниченко Д. Linux. От новичка к профессионалу. 2-е изд. СПб: БХВ-Петербург, 2010. 764 с.
7. Crist E.F., Keijser J.J. Mastering OpenVPN. Birmingham – Mumbai, Puckt Publ., 2015, 364 p.
8. Ватаманюк А.К. Создание, обслуживание и администрирование сетей на 100 %. СПб: Питер, 2010. 350 с.
9. Dubrova E. Fault-Tolerant Design. Springer, 2013, 185 p.
10. Pargo C., Стивенс У. UNIX. Профессиональное программирование, 3-е изд. СПб, 2013. 1104 с.
11. Krylov V., Kravtsov K. DDoS attack and interception resistance IP fast hopping based protocol: 23rd Intern. Conf. on Software Engineering and Data Engineering, 2014, pp. 43–48.
12. Borodin A., Chernyshov A. The study of the network stack under the influence of static load. In the world of scientific discovery. 2014, vol. 60, no. 12.2, pp. 601–612.
13. Блум Р., Бреснахэн К. Командная строка Linux и сценарии оболочки. М.: Вильямс, 2013. 784 с.
14. Сокольников А.М. Сравнительный анализ подходов к разработке архитектуры и систем управления базами данных для высоконагруженных web-сервисов // Кибернетика и программирование. 2014. № 4. С. 1–13.
15. Leu F.-Y., Lin I.-L. A DoS/DDoS attack detection system using chi-square statistic approach. Jour. of Systemics, Cybernetics and Informatics. 2010, vol. 8, no. 2, pp. 41–51.

AUTOMATED DATA PROCESSING SYSTEM IN UNIX-LIKE SYSTEMS

E.V. Palchevsky¹, *Postgraduate Student, teelxp@inbox.ru*

A.R. Khalikov¹, *Ph.D. (Physics and Mathematics), Associate Professor, khalikov.albert.r@gmail.com*

¹ *Ufa State Aviation Technical University, K. Marks St. 12, Ufa, 450000, Russian Federation*

Abstract. The article considers development of distributed and modular information processing in the automated mode. Implementation of this project allows accepting input and output data on a physical server up to 2,2 GB/s in volume while distributing stream information (all incoming network traffic on the server) on physical and logical kernels.

The paper shows a load dependence of physical resources on the input information. It also justifies feasibility of the developed hardware and software SDP (Speed data processing) use. There is also a structure and basic operation diagrams. The first stage of creating a complex is the development of an algorithm. The second stage includes its technical implementation. The article gives a fragment of a source code, which is responsible for notification messages on an email address as CPU load, and the main launched processes. There is a description of the main functionality with the following data: a function name, a function purpose, theoretical loading, data transfer limit (in MB/s) and a result of execution. The third stage includes testing SDP complex. Its average daily results are provided within ten days.

The created hardware and software complex allows processing input and output information effectively in an automatic mode in order to increase capacity of reception and sending data in MySQL DBMS, including cases with DoS- and the DDoS-attacks. One of the parts of the complex is WEB-module, which is responsible for control, both as from a personal computer, as well as from a mobile phone. The possibility of notifying about a physical server usage by sms is realized in a monitoring part of the WEB module.

The developed hardware and software complex showed high stability when handling data in large volumes with a minimum load on a computer.

Keywords: information processing, load on computing resources, data structure, lowering of information flow loading, information flows, SMS notification, distributed systems, DPC, data processing center.

References

1. Palchevsky E.V., Khalikov A.R. TCP/IP network STACK optimization under high load on UNIX-like systems. *Proc. DSPTech '2015*. UFA, USATU Publ., 2015, vol. 1, pp. 130–135.
2. Palchevsky E.V., Khalikov A.R. Technique the Instrumentation A Code And Optimization of Code Lines in Modeling Phase Transitions on the Programming Language C++. *Proc. Institute for system Programming of the RAS*. 2015, vol. 27, iss. 6, pp. 87–96 (in Russ.).
3. Palchevsky E.V., Khalikov A.R. Uniformly distributed load of hardware and software core in the UNIX-based systems. *Proc. Institute for system Programming of the RAS*. 2016, vol. 28, iss. 1, pp. 93–102 (in Russ.).
4. *IBM A Management System for the Information Business*. White Plains, NY, IBM Publ., 1980.
5. Olifer V.G., Olifer N.A. *Computer networks. Principles, technologies, protocols*. St. Petersburg, 2007, 960 p.
6. Kolisnichenko D. *Linux. Ot novichka k professionalu* [Linux. From a Beginner to a Professional]. 2nd ed., St. Petersburg, BHV-Peterburg Publ., 2010, 764 p.
7. Crist E.F., Keijser J.J. *Mastering OpenVPN*. Birmingham – Mumbai, Puckt Publ., 2015, 364 p.
8. Vatamanyuk A.K. *Sozдание, obsluzhivanie i administrirovanie setey na 100 %* [100 % Creation, Service and Administration of Networks]. St. Petersburg, Piter Publ., 2010, 350 p.
9. Dubrova E. *Fault-Tolerant Design*. Springer Publ., 2013, 185 p.
10. Rago S.A., Stevens W.R. *Advanced Programming in the UNIX Environment*. 3rd ed., Addison-Wesley Professional Publ., 2013, 1024 p. (Russ. ed.: St. Petersburg, 2013, 1104 p.).
11. Krylov V., Kravtsov K. DDoS attack and interception resistance IP fast hopping based protocol. *Proc. 23rd Int. Conf. on Software Engineering and Data Engineering*. 2014, pp. 43–48.
12. Borodin A., Chernyshov A. The study of the network stack under the influence of static load. *V mire nauchnykh otkryty* [In the World of Scientific Discoveries]. 2014, vol. 60, no. 12.2, pp. 601–612 (in Russ.).
13. Blum R., Bresnakhen K. *Linux Command Line and Shell Scripting Bible*. Wiley Publ., 816 p. (Russ.ed.: Villiams Publ., 2013, 784 p.).
14. Sokolnikov A.M. Comparative analysis of the approaches in development of the database management systems and its' architecture for highly loaded web-services. *Kibernetika i programmirovaniye* [Cybernetics and programming]. 2014, no. 4, pp. 1–13 (in Russ.).
15. Leu F.-Y., Lin I.-L. A DoS/DDoS Attack Detection System Using Chi-Square Statistic Approach. *Jour. of Systemics, Cybernetics and Informatics*. 2010, vol. 8, no. 2, pp. 41–51.

УДК 517.939

DOI: 10.15827/0236-235X.118.235-240

Дата подачи статьи: 31.03.17

2017. Т. 30. № 2. С. 235–240

ОБОБЩЕННО-ПЕРИОДИЧЕСКИЕ ДВИЖЕНИЯ ДИНАМИЧЕСКИХ И НЕАВТОНОМНЫХ ПЕРИОДИЧЕСКИХ СИСТЕМ

А.П. Афанасьев, д.ф.-м.н., профессор, ara@isa.ru

(Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН),

Большой Каретный пер., 19, стр. 1, г. Москва, 127051, Россия;

Московский государственный университет им. М.В. Ломоносова,

Ленинские горы, 1, г. Москва, 119991, Россия;

Национальный исследовательский технологический университет «МИСиС»,

Ленинский просп., 6, г. Москва, 119049, Россия;

НИУ «Высшая школа экономики», ул. Мясницкая, 20, г. Москва, 101000, Россия);

С.М. Дзюба, д.ф.-м.н., профессор, sdzyuba@mail.ru;

И.И. Емельянова, ст. преподаватель, emelyanova-123@yandex.ru

(Тверской государственный технический университет,

наб. Аф. Никитина, 22, г. Тверь, 170026, Россия)

Настоящий обзор посвящен истории изучения ситуации общего положения движений динамических и неавтономных периодических систем, поскольку, как оказалось, полное и подробное описание ситуации общего положения позволяет решить проблему построения обобщенно-периодических движений динамических и неавтономных периодических систем. Необходимость численного изучения указанных систем объясняется тем, что подавляющее большинство моделей реальных технических, биологических, экономических и др. процессов описываются именно такими системами.

В автономном случае обобщенно-периодическое движение эквивалентно классическому рекуррентному движению, введенному и изученному еще Дж. Биркгофом. С понятием рекуррентного движения прямо связано понятие минимального множества. Именно эти два понятия и определяют ситуацию общего положения классических динамических систем.

Особое значение построение и исследование рекуррентных движений и минимальных множеств приобрели в связи с потребностями хаотической динамики и гиперболической теории. Однако еще до недавнего времени отсутствовали общие методы построения рекуррентных движений и минимальных множеств: все сводилось к построению аттракторов отдельных систем дифференциальных уравнений с полилинейной правой частью. Открытие же понятия обобщенно-периодического движения привело к созданию общего метода построения и исследования всех минимальных множеств, содержащихся в предельных множествах динамических систем. Более того, понятие обобщенно-периодического движения позволило перенести все основные понятия классической теории динамических систем на неавтономные периодические системы и описать ситуацию общего положения в таких системах с единых позиций. Соответственно, появилась возможность прямого переноса метода построения рекуррентных движений на построение обобщенно-периодических движений неавтономных периодических систем, что позволило выполнять численное построение таких движений с единых позиций.

Ключевые слова: динамические и неавтономные периодические системы, ситуация общего положения, обобщенно-периодические движения, построение обобщенно-периодических движений.

До недавнего времени отсутствовали какие-либо общие методы и алгоритмы построения рекуррентных решений и минимальных множеств автономных систем обыкновенных дифференциальных уравнений. Как оказалось, причина в том, что ситуация общего положения решений не была достаточно изучена.

Что же мы понимаем под ситуацией общего положения? Для ответа на этот вопрос рассмотрим нормальную систему дифференциальных уравнений, векторная запись которой имеет вид

$$x' = f(t, x), \quad (1)$$

где $x = (x^1, \dots, x^n)$ – векторная функция действительного переменного t , а $f = (f^1, \dots, f^n)$ – векторная функция, определенная и непрерывная вместе со

своими частными производными $\frac{\partial f^i}{\partial x^j}$, $i, j = 1, \dots, n$,

на прямом произведении $R \times R^n$ действительной оси R и евклидова векторного пространства R^n .

Кроме того, предположим, что f – T -периодическая по t функция.

Если $n = 2$, то справедлива следующая теорема Х.Л. Массера [1]: пусть каждое решение $\xi(t)$ системы (1) определено для всех $t \geq t_0$ ($t \leq t_0$); тогда, если (1) имеет решение, ограниченное при этих значениях t , данная система имеет T -периодическое решение $\varphi(t)$. Согласно [1], это утверждение справедливо и для линейных систем произвольного порядка. Однако в нелинейном случае теорема Массера неверна уже при $n = 3$ [2].

Задолго до появления теоремы Массера для гладких автономных систем

$$\dot{x} = f(x) \quad (2)$$

Дж. Биркгофом было введено определение рекуррентного решения. Данное решение содержит в себе как частный случай периодическое решение. При этом из существования ограниченного решения всегда следует существование рекуррентного

решения и наоборот. Именно эту ситуацию мы и называем ситуацией общего положения решений автономных систем.

Свойства решений системы (1), вообще говоря, несколько отличаются от свойств решений системы (2). Так, например, в неавтономном случае траектории решений могут пересекаться. Это приводит к тому, что определение рекуррентности на неавтономный случай прямо не переносится. Отсюда следует, что для неавтономной системы вида (1) при $n > 2$ ситуация общего положения периодическим решением не определена, а рекуррентным решением не определяется (ввиду отсутствия определения последнего).

Исследования, выполненные в работах [3–5], привели к определению обобщенно-периодического решения [2]. Оказалось, что из существования ограниченного решения всегда следует существование обобщенно-периодического решения. Более того, в автономном случае это определение эквивалентно определению рекуррентного решения. Другими словами, обобщенно-периодическое решение дает ситуацию общего положения и в автономном, и в неавтономном случаях.

Выясним историческую роль и место ситуации общего положения в теории динамических систем.

Хорошо известно, что геометрическое исследование А. Пуанкаре «О кривых, определяемых дифференциальными уравнениями» [6], а также его знаменитый мемуар «Новые методы небесной механики» [7, 8] в сочетании с диссертацией А.М. Ляпунова [9] заложили в конце девятнадцатого века основы качественной теории дифференциальных уравнений.

На взгляд авторов, особого упоминания заслуживает мемуар А. Пуанкаре «Об одной геометрической теореме» – прощальный поклон и завещание великого ученого [8]. Сформулированная здесь в виде гипотезы теорема о неподвижной точке некоторого гладкого отображения кругового кольца в себя была доказана Дж. Биркгофом уже после смерти А. Пуанкаре. Трудно утверждать наверняка, но, возможно, именно работа над этой теоремой окончательно укрепила силы и дух Дж. Биркгофа и привела его к дальнейшим исследованиям по теории динамических систем и топологии. Во всяком случае современники утверждали, что «настоящим учителем Биркгофа был Пуанкаре» и что «Биркгоф усердно изучал все работы Пуанкаре по динамике» (например [8]).

Каково же отношение Дж. Биркгофа к общей теории динамических систем? Ответ на этот вопрос прост – он ее создал.

В основе общей теории динамических систем лежит понятие движения. В случае автономной системы (2) движение эквивалентно решению, определенному на всей оси R . В случае абстрактной динамической системы, характеризуемой однопараметрической группой преобразований, движение

естественным образом обобщает понятие решения.

Конечной целью общей теории динамических систем является определение всех возможных типов движений и установление связи между ними. В [10] введены и подробно изучены такие фундаментальные понятия, как α - и ω -предельные точки и множества, возвращаемость областей и центральные движения, минимальные множества и рекуррентные движения. Свои результаты Биркгоф дополнил знаменитой эргодической теоремой, заложив, таким образом, основы эргодической теории.

Нельзя сказать, что книга Дж. Биркгофа написана простым и понятным языком, особенно для своего времени: об этом в своих воспоминаниях говорил даже Л.С. Понтрягин. Однако уже в середине прошлого века она нашла последователей и получила развитие в работах А.А. Маркова, Г.Ф. Хильми, В.В. Немыцкого, В.В. Степанова, А.Н. Колмогорова и многих других математиков (например в [11]).

Наряду с работами по общей теории динамических систем качественная теория дифференциальных уравнений одновременно получила более чем продвинутое развитие в области систем с интегральным инвариантом. Полученные в этом направлении результаты К. Каратеодори и А.Я. Хинчина существенным образом дополнили и обобщили теорему Пуанкаре о возвращении.

Наряду с упомянутыми работами в качественной теории дифференциальных уравнений появилось и совершенно новое важное направление – структурная устойчивость, изначально названная грубостью динамических систем. Основоположниками в этой области выступили А.А. Андронов и Л.С. Понтрягин, создавшие теорию грубых систем на плоскости [12–15]. Ими, в частности, было доказано, что грубые системы всюду плотны на множестве двумерных систем.

Перенесение понятия грубости уже на трехмерный случай оказалось существенно осложненным. Объясняется это гораздо менее тривиальным характером фазовых портретов грубых трехмерных систем. Принципиально важным оказалось то, что здесь впервые был построен пример грубого диффеоморфизма со счетным множеством седловых предельных циклов, имеющих неограниченно возрастающий период [16].

Существенным шагом вперед в этом (и не только в этом) направлении стала книга [17] (см. также [18]). В ней наиболее привлекают не установленная структурная устойчивость геодезических потоков на замкнутых римановых многообразиях отрицательной кривизны, а четкое описание вида и установление свойств динамических и дискретных систем, названных впоследствии У-системами Д.В. Аносова.

Аносов показал, что каждая У-система структурно устойчива. Более того, оказалось, что структурная устойчивость таких систем неизбежно соче-

тается с экспоненциальным разбеганием фазовых кривых и перемешиванием.

Особый интерес к этому открытию подогрела появившаяся в начале 60-х годов прошлого века следующая попытка построения теории турбулентного движения жидкости, непосредственно приводящая к понятию гиперболического притягивающего множества (например [18]).

Следуя [18], рассмотрим замкнутый сосуд, заполненный несжимаемой вязкой жидкостью, приводимой в движение внешней силой – мешалкой. Динамику движения жидкости описывают уравнениями Навье–Стокса. Точками фазового пространства такой системы являются бездивергентные векторные поля скоростей жидкости, в которых положения равновесия представляют собой стационарные поля скоростей. Циклы же здесь отождествляют с периодическими движениями жидкости, в которых скорость движения в каждой точке пространства меняется по периодическому закону.

Если вязкость жидкости достаточно велика, то система уравнений Навье–Стокса имеет единственное положение равновесия, к которому притягиваются все фазовые кривые. Такое движение жидкости называют ламинарным, и всякое другое течение под действием вязкости со временем стремится к нему.

С уменьшением вязкости ламинарное движение может потерять устойчивость, что может привести к образованию структурно устойчивого предельного цикла. Дальнейшее уменьшение вязкости, вообще говоря, может вызвать потерю устойчивости данного цикла. При этом возможно появление из цикла некоторого другого, более сложного непериодического движения, притягивающего соседние.

Гипотеза математического описания турбулентности состоит в том, что в бесконечномерном фазовом пространстве имеется конечное множество, к которому притягиваются все фазовые кривые. Само же притягивающее множество должно представлять собой фазовое пространство или U -системы, или системы, качественно близкой к ней: в любом случае здесь должны сохраняться экспоненциальное разбегание траекторий и следующее из него перемешивание.

С физической точки зрения данная гипотеза выглядит весьма правдоподобно и подтверждается многочисленными экспериментальными исследованиями. Однако, несмотря на все прилагаемые усилия превратить ее в теорему, насколько авторам известно, в настоящее время так и не удалось.

Оставив в стороне вопросы структурной устойчивости, заметим, что с этого времени взгляд на теорию динамических систем постепенно начал претерпевать изменения. В наступившей эпохе гипермодернизма понятия рекуррентного движения и минимального множества постепенно стали исчезать из употребления, а о ситуации общего положе-

ния вообще перестали говорить. Так, например, попытка отыскать аналог рекуррентного движения для неавтономных систем фактически сводилась к изучению траекторий дискретных динамических систем [19].

В работах [20, 21] для изучения ситуации общего положения в абстрактных автономных динамических и неавтономных периодических системах был предложен метод, существенным образом использующий только групповые свойства оператора сдвига вдоль движений этих систем. В основе данного метода, собственно, и лежит определение движения, которое будем называть обобщенно-периодическим.

Существование в любой из указанных систем движения при $t \geq t_0$, расположенного в некотором компактном множестве, влечет за собой существование обобщенно-периодического движения. Оказалось, что в автономном случае определение равномерно обобщенно-периодического движения эквивалентно определению рекуррентного движения. При этом определение равномерно устойчивого движения наиболее точно передает характер последнего и, кроме того, устанавливает обязательный (причем достаточно жесткий) сценарий асимптотического перехода произвольного движения к рекуррентному. Последнее позволило в работах [22, 23] осуществить разработку общего метода построения рекуррентных движений и минимальных множеств классических динамических систем. Более того, как показано в работе [24], данный метод без каких-либо существенных изменений переносится на проблему построения обобщенно-периодических движений неавтономных периодических систем.

Заметим, что изучение ситуации общего положения и введение понятия обобщенно-периодического движения дают новые характеристические свойства рекуррентных движений, дополняющие классические [22, 24]. Эти новые свойства дают ряд дополнительных возможностей для исследования динамических систем. Проиллюстрируем их на следующих примерах.

Пример 1. Важнейшими из всех качественных типов движений, как известно, являются устойчивые по Пуассону и рекуррентные движения. Отличительной особенностью этих движений является то, что каждое рекуррентное движение устойчиво по Пуассону, но не наоборот. При этом из существования движения, положительно или отрицательно устойчивого по Лагранжу, следует существование, по крайней мере, одного рекуррентного движения. Однако из существования рекуррентного движения не следует существования устойчивого по Пуассону, но не рекуррентного движения. В настоящее время связь между рекуррентными и устойчивыми по Пуассону нерекуррентными движениями изучена на уровне определений [25], а разработанная в [3–5, 20–24] техника позволяет

осуществить более полное установление такой связи.

Напомним, что множество M называется минимальным, если оно непусто, замкнуто, инвариантно и не содержит ни одного собственного подмножества, обладающего этими тремя свойствами. Заметим также, что в компактном метрическом пространстве движение замыкание траектории рекуррентного движения является компактным минимальным множеством M и наоборот. При этом замыкание H траектории устойчивого по Пуассону движения называется квазiminимальным множеством.

Каждое квазiminимальное множество H компактно и содержит компактное минимальное множество M . Однако M не обязано быть собственным подмножеством множества H . Именно случай, в котором M является собственным подмножеством множества H , и определяет ситуацию существования устойчивого по Пуассону нерекуррентного движения.

Как показано в работе [22], в квазiminимальном множестве H , не являющемся минимальным множеством, объединение всех минимальных множеств компактно и нигде не плотно в H . При этом точки, лежащие на траекториях нерекуррентных устойчивых по Пуассону движений, образуют множество второй категории типа G_δ , то есть его дополнение может быть представлено как объединение счетного числа замкнутых множеств, нигде не плотных в H . Более полно, согласно теореме Хильми, в квазiminимальном множестве H , не являющемся минимальным множеством, точки, лежащие на траекториях нерекуррентных устойчивых по Пуассону движений и всюду плотных в H , образуют множество второй категории типа G_δ . Несложно также показать, что каждое такое квазiminимальное множество H содержит несчетное множество нерекуррентных устойчивых по Пуассону движений, всюду плотных в H . Как оказалось, все это имеет большое значение при численном исследовании уравнений хаотической динамики, в частности, при построении странных аттракторов (см. пример 3).

Пример 2. Системы с инвариантной мерой, как известно, обладают рядом свойств, выгодно выделяющих их из всего множества динамических систем. Важнейшее из этих свойств состоит в том, что, согласно теореме о возвращении точек Пуанкаре–Каратеодори, для систем, имеющих в сепарабельном пространстве положительную конечную инвариантную меру, почти все точки пространства устойчивы по Пуассону. Как показано в монографии [2], применение метода, использовавшегося при описании ситуации общего положения, позволяет существенно уточнить этот классический результат. Именно в условиях теоремы Пуанкаре–Каратеодори каждая устойчивая по Пуассону точка является точкой какого-либо минимального мно-

жества, то есть почти все точки пространства являются точками минимальных множеств.

Пример 3. В последнее время при исследовании уравнений хаотической динамики стали широко применяться численные методы. В частности, почти все известные на сегодняшний день странные аттракторы построены именно численно.

Для определенности рассмотрим аттрактор Лоренца [14, 23]. Согласно многочисленным вычислительным экспериментам многих исследователей, данный аттрактор состоит из трех положений равновесия (один седлоузел и два седлофокуса) и множества незамкнутых устойчивых по Пуассону траекторий, всюду плотно заполняющих этот аттрактор (за исключением седлофокусов). Более того, сам аттрактор представляет собой объединение трех положений равновесия и замыкания устойчивой по Пуассону траектории. При этом считается также, что аттрактор содержит счетное всюду плотное множество седловых предельных циклов [14]. Проведенные авторами численные эксперименты показали, что данная гипотеза нуждается в существенной доработке.

Действительно, как отмечалось в примере 1, в квазiminимальном множестве H , не являющемся минимальным множеством, объединение всех минимальных множеств компактно и нигде не плотно. Поэтому, если аттрактор Лоренца имеет указанную выше структуру относительно положений равновесия и устойчивых по Пуассону траекторий, он не может содержать счетного всюду плотного множества седловых предельных циклов. Более того, как показано в [23], с точностью до ошибки вычислений можно утверждать, что в системе Лоренца имеются лишь три указанных выше положения равновесия и минимальное множество, образованное незамкнутым рекуррентным движением. Все эти данные, полученные специально разработанным численным методом, хорошо согласуются с теоретическими результатами, приведенными в [22, 24].

Литература

1. Massera J.L. The Existence of Periodic Solutions of Systems of Differential Equations. Duke Math. J., 1950, vol. 17, pp. 457–475.
2. Афанасьев А.П., Дзюба С.М. Устойчивость по Пуассону в динамических и непрерывных периодических системах. М.: ЛКИ, 2007. 240 с.
3. Афанасьев А.П., Дзюба С.М. К вопросам управления в периодических процессах // Изв. РАН. Теория и системы управления. 1998. № 4. С. 15–20.
4. Дзюба С.М. Два многомерных дополнения к теореме Пуанкаре–Бендиксона // Дифференц. уравнения. 1995. Т. 31. № 4. С. 579–582.
5. Дзюба С.М. Об условно-периодических решениях дифференциальных уравнений // Дифференц. уравнения. 1999. Т. 35. № 8. С. 1020–1023.
6. Пуанкаре А. О кривых, определяемых дифференциальными уравнениями. М.–Л.: ОГИЗ, 1947. 256 с.
7. Пуанкаре А. Избранные научные труды. М.: Наука, 1971. Т. 1. 772 с.

8. Пуанкаре А. Избранные научные труды. М.: Наука, 1972. Т. 2. 1000 с.
9. Ляпунов А.М. Общая задача об устойчивости движения. М.: Изд-во ОНТИ, 1935. 174 с.
10. Биркгоф Дж. Динамические системы. М.–Л.: ОГИЗ, 1941. 332 с.
11. Немыцкий В.В., Степанов В.В. Качественная теория дифференциальных уравнений. М.: Едиториал УРСС, 2004. 550 с.
12. Андронов А.А., Витт А.А., Хайкин С.Э. Теория колебаний. М.: Физматгиз, 1959. 550 с.
13. Андронов А.А., Понтрягин Л.С. Грубые системы // Докл. АН СССР. 1937. Т. 14. № 5. С. 247–251.
14. Баутин И.Н., Леонтович Е.А. Методы и приемы качественного исследования динамических систем на плоскости. М.: Наука, 1990. 486 с.
15. Понтрягин Л.С. Избранные научные труды. М.: Наука, 1988. Т. 2. 576 с.
16. Смейл С. Структурно устойчивый дифференцируемый гомеоморфизм с бесконечным числом периодических точек. Киев: Изд-во Ин-та матем. АН УССР, 1961. 426 с.
17. Аносов Д.В. Геодезические потоки на замкнутых римановых многообразиях отрицательной кривизны // Тр. МИАН. 1967. № 9. С. 3–24.
18. Арнольд В.И. Дополнительные главы теории обыкновенных дифференциальных уравнений. М.: Наука, 1978. 304 с.
19. Хейл Дж. Теория функционально-дифференциальных уравнений. М.: Мир, 1984. 421 с.
20. Афанасьев А.П., Дзюба С.М. Периодический оператор сдвига и квазипериодические кривые // Дифференц. уравнения. 2004. Т. 40. № 10. С. 1367–1372.
21. Афанасьев А.П., Дзюба С.М. О рекуррентных траекториях, минимальных множествах и квазипериодических движениях динамических систем // Дифференц. уравнения. 2005. Т. 41. № 11. С. 1469–1474.
22. Афанасьев А.П., Дзюба С.М. Метод построения минимальных множеств динамических систем // Дифференц. уравнения. 2015. Т. 51. № 7. С. 835–841.
23. Афанасьев А.П., Дзюба С.М. Построение минимальных множеств динамических систем полиномиальной правой частью // Дифференц. уравнения. 2015. Т. 51. № 11. С. 1411–1419.
24. Афанасьев А.П., Дзюба С.М. Обобщенно-периодические движения неавтономных систем // Дифференц. уравнения. 2017. Т. 53. № 1. С. 3–9.
25. Cheban D.N. Asymptotically almost periodic solutions of differential equations, HPC, NY, 2009, 204 p.

Software & Systems

DOI: 10.15827/0236-235X.118.235-240

Received 31.03.17

2017, vol. 30, no. 2, pp. 235–240

GENERALIZED PERIODIC MOTIONS OF DYNAMICAL AND NON-AUTONOMOUS PERIODIC SYSTEMS

A.P. Afanasiev^{1,2,3,4}, Dr.Sc. (Physics and Mathematics), Professor, apa@isa.ruS.M. Dzyuba⁵, Dr.Sc. (Physics and Mathematics), Professor, sdzyuba@mail.ruI.I. Emelyanova⁵, Senior Lecturer, emelyanova-123@yandex.ru¹ Institute for Information Transmission Problems of the Russian Academy of Sciences, IITP RAS, Bolshoy Karetny Lane 19, bldg. 1, Moscow, 127994, Russian Federation² Lomonosov Moscow State University, Leninskie Gory, Moscow, 119991, Russian Federation³ National University of Science Research Technological "MISIS",

Leninsky Ave. 6, Moscow, 119049, Russian Federation

⁴ National Research University, Higher School of Economics, Myasnitskaya St. 20, Moscow, 101000, Russian Federation⁵ Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation

Abstract. This review represents the history of study analysis of the situation of typical behavior of dynamical and non-autonomous periodic system motions. The necessity of this investigation is due to the fact that the complete and detailed description of a typical behavior situation makes it possible to solve the problem of construction of generalized periodic motions of dynamical and non-autonomous periodic systems. The necessity of numerical investigation of indicated systems is due to the fact that the vast majority of real technological, biological, economical and others process models are described by these specific systems.

In the autonomous case a generalized periodic motion is equivalent to a classical recurrent motion, which was introduced and investigated by G. Birkhoff. The concept of the recurrent motion is related to the concept of a minimal set. These two concepts define the situation of typical behavior of classical dynamical systems.

The construction and investigation of recurrent motions and minimal sets took on special significance due to the requirements of chaotic dynamics and the hyperbolic theory. However, until quite recently there were no general methods of constructing recurrent motions and minimal sets. It came down to constructing attractors of separate systems of differential equations with a polylinear right part. The discovery of the generalized periodic motion concept resulted in creation of the general method of construction and investigation of all minimal sets contained in limit sets of dynamical systems. Furthermore, the concept of a generalized periodic motion allowed adapting all the basic definitions of the classical dynamical system theory to non-autonomous periodic systems and describing the situation of typical behavior in such systems on common ground. Therefore, it allows adapting directly the method of recurrent motion construction to the construction of generalized periodic motions of non-autonomous periodic systems that made the numerical construction of such motions on common ground possible.

Keywords: dynamical and non-autonomous periodic systems, situation of typical behavior, generalized-periodic motions, construction of generalized-periodic motions.

References

1. Massera J.L. The Existence of Periodic Solutions of Systems of Differential Equations. *Duke Math. J.* 1950, vol. 17, pp. 457–475.
2. Afanasiev A.P., Dzyuba S.M. *Ustoychivost po Puassonu v dinamicheskikh i nepreryvnykh periodicheskikh sistemakh* [Poisson stability in dynamic and indiscrete periodic systems]. Moscow, LKI Publ., 2007, 240 p.
3. Afanasiev A.P., Dzyuba S.M. On the management question in periodic processes. *Izv. RAN. Teoriya i sistemy upravleniya* [Jour. of Computer and Systems Sciences Int.]. 1998, no. 4, pp. 15–20 (in Russ.).
4. Dzyuba S.M. Two multidimensional supplements to the Poincaré–Bendixson theorem. *Differents. uravneniya* [Differential Equations]. 1995, vol. 31, no. 4, pp. 579–582 (in Russ.).
5. Dzyuba S.M. On conditionally periodic solutions of differential equations. *Differents. uravneniya* [Differential Equations]. 1999, vol. 35, no. 8, pp. 1020–1023 (in Russ.).
6. Poincaré J.H. *O krivykh, opredelyaemykh differentsialnymi uravneniyami* [On Curves That are Defined by Differential Equations]. Moscow, L., OGIZ Publ., 1947, 256 p.
7. Poincaré J.H. *Izbrannye nauchnye trudy* [Selected Scientific Papers]. Moscow, Nauka Publ., 1971, vol. 1, 772 p.
8. Poincaré J.H. *Izbrannye nauchnye trudy* [Selected Scientific Papers]. Moscow, Nauka Publ., 1972, vol. 2, 1000 p.
9. Lyapunov A.M. *Obshchaya zadacha ob ustoychivosti dvizheniya* [A General Dynamical Stability Problem]. Moscow, ONTI Publ., 1935, 174 p.
10. Birkhoff G.D. *Dinamicheskie sistemy* [Dynamic System]. Moscow, L., OGIZ Publ., 1941, 332 p.
11. Nemytsky V.V., Stepanov V.V. *Kachestvennaya teoriya differentsialnykh uravneniy* [Qualitative Theory of Differential Equations]. Moscow, Editorial URSS Publ., 2004, 550 p.
12. Andronov A.A., Vitt A.A., Khaykin S.E. *Teoriya kolebaniy* [Oscillation Theory]. Moscow, Fizmatgiz Publ., 1959, 550 p.
13. Andronov A.A., Pontryagin L.S. Coarse systems. *Dokl. AN SSSR* [Proc. USSR Science Academy]. 1937, vol. 14, no. 5, pp. 247–251 (in Russ.).
14. Bautin I.N., Leontovich E.A. *Metody i priemy kachestvennogo issledovaniya dinamicheskikh sistem na ploskosti* [Methods and Procedures of Qualitative Research on Dynamic Systems on a Plane]. Moscow, Nauka Publ., 1990, 486 p.
15. Pontryagin L.S. *Izbrannye nauchnye trudy* [Selected Scientific Papers]. Moscow, Nauka Publ., 1988, vol. 2, 576 p.
16. Smeyl S. *Strukturno ustoychivy differentsiruemy gomeomorfizm s beskonechnym chislom periodicheskikh toчек* [Structurally Stable Differentiable Homeomorphism with an Infinite Number of Periodic Points]. Kiev, In-t matematiki AN USSR Publ., 1961, 426 p.
17. Anosov D.V. *Geodezicheskie potoki na zamknutykh rimanovykh mnogoobraziyakh otritsatel'noy krivizny* [Geodesic Flows on Closure Riemann's Manifolds of Negative Curvature]. Moscow, Tr. MIAN Publ., 1967, no. 9, pp. 3–24 (in Russ.).
18. Arnold V.I. *Dopolnitelnye glavy teorii obyknovennykh differentsialnykh uravneniy* [Additional Chapters of the Ordinary Differential Equation Theory]. Moscow, Nauka Publ., 1978, 304 p.
19. Hale J. *Theory of Functional Differential Equations*. Springer Publ., 1977 (Russ. ed.: Moscow, Mir Publ., 1984, 421 p.).
20. Afanasiev A.P., Dzyuba S.M. Periodic translation operators and quasiperiodic curves. *Differents. uravneniya* [Differential Equations]. 2004, vol. 40, no. 10, pp. 1367–1372 (in Russ.).
21. Afanasiev A.P., Dzyuba S.M. On recurrent trajectories, minimal sets, and quasiperiodic motions of dynamical systems. *Differents. uravneniya* [Differential Equations]. 2005, vol. 41, no. 10, pp. 1469–1474.
22. Afanasiev A.P., Dzyuba S.M. Method for constructing minimal sets of dynamical systems. *Differents. uravneniya* [Differential Equations]. 2015, vol. 51, no. 7, pp. 835–841 (in Russ.).
23. Afanasiev A.P., Dzyuba S.M. Construction of the minimal sets of differential equations with polynomial right-hand side. *Differents. uravneniya* [Differential Equations]. 2015, vol. 51, no. 11, pp. 1411–1419 (in Russ.).
24. Afanasiev A.P., Dzyuba S.M. Generalized-periodic motions of nonautonomous systems. *Differents. uravneniya* [Differential Equations]. 2017, vol. 53, no. 1, pp. 3–9 (in Russ.).
25. Cheban D.N. *Asymptotically Almost Periodic Solutions of Differential Equations*. HPC, NY, 2009, 204 p.

Примеры библиографического описания статьи

1. Афанасьев А.П., Дзюба С.М., Емельянова И.И. Обобщенно-периодические движения динамических и неавтономных периодических систем // Программные продукты и системы. 2017. Т. 30. № 2. С. 235–240; DOI: 10.15827/0236-235X.118.235-240.
2. Afanasiev A.P., Dzyuba S.M., Emelyanova I.I. Generalized periodic motions of dynamical and non-autonomous periodic systems. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 235–240 (in Russ.); DOI: 10.15827/0236-235X.118.235-240.

УДК 681.322

DOI: 10.15827/0236-235X.118.241-256

Дата подачи статьи: 08.12.16

2017. Т. 30. № 2. С. 241–256

ОЦЕНКА ЭФФЕКТИВНОСТИ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ОБЕСПЕЧЕНИЯ УСТОЙЧИВОСТИ ФУНКЦИОНИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Д.О. Есиков аспирант, mcgeen4@gmail.com

(Тульский государственный университет, просп. Ленина, 92, г. Тула, 300012, Россия)

Для принятия обоснованных решений в части организации процессов хранения и обработки данных для обеспечения устойчивости функционирования распределенных информационных систем предложено применять комплекс разработанных математических моделей оптимизации распределения элементов ПО функциональных задач по узлам сети, распределения информационных ресурсов по центрам хранения и обработки данных, состава технических средств системы хранения и обработки данных, распределения резерва информационных ресурсов по центрам хранения и обработки данных. Показано, что данные задачи относятся к классу задач дискретной оптимизации с булевыми переменными.

Для решения формализованных задач предложено и экспериментально проверено применение метода ветвей и границ и генетических алгоритмов. Для повышения эффективности этого метода рекомендован алгоритм предварительного определения порядка ветвления переменных путем однократного решения приближенным методом задачи, двойственной по отношению к исходной. Проведена экспериментальная проверка эффективности метода ветвей и границ для решения задач обеспечения устойчивости функционирования распределенных информационных систем, в том числе с использованием алгоритма предварительного определения порядка ветвления переменных. Дана оценка влияния исходных данных на общую производительность метода ветвей и границ. Определены наиболее эффективные для решения разработанных задач стратегии ветвления переменных. Предложены варианты основных операторов, а также схемы начальной инициализации исходной популяции генетического алгоритма для решения задач обеспечения устойчивости функционирования распределенных информационных систем.

Для повышения качества получаемого генетическим алгоритмом решения обосновано использование адаптивной схемы репродукции особей и островной схемы организации вычислений. Экспериментально проверена эффективность предложенных генетического и островного генетического алгоритмов и определены параметры генетических алгоритмов, обеспечивающие максимальное качество получаемого решения. Подтверждена возможность управления точностью получаемого решения за счет изменения параметров алгоритма при введении ограничений на время решения. Дана сравнительная оценка метода ветвей и границ и островного генетического алгоритма при решении формализованных задач, определены области их эффективного применения.

Ключевые слова: *распределенная информационная система, устойчивость функционирования, математические модели, метод ветвей и границ, двойственная задача, генетический алгоритм, островной генетический алгоритм, адаптивная схема репродукции.*

Устойчивое функционирование *распределенных информационных систем* (РИС) является залогом эффективной работы организаций и предприятий в различных сферах экономики. Под устойчивостью функционирования системы понимается ее способность выполнять возложенные функции с заданными показателями качества в условиях воздействия внутренних и внешних дестабилизирующих факторов.

В отличие от надежности под устойчивостью подразумевается обеспечение требуемых значений показателей эффективности системы в условиях возможной ее реконфигурации (изменения ее структуры).

В общем случае дестабилизирующие факторы в РИС делятся по местоположению источника на внешние и внутренние, а по характеру возникновения – на случайные и преднамеренные [1].

Наиболее опасными с точки зрения обеспечения целевых показателей функционирования РИС являются дестабилизирующие факторы, в результате воздействия которых ухудшаются параметры процессов обеспечения информационных запросов пользователей, хранения и обработки информации.

К таким факторам можно отнести следующие:

- структурная деградация сети (выход из строя средств обработки информации в узлах распределенной системы, каналов передачи информации);
- разрушение элементов ПО (разрушение данных и кодов исполняемых программ, внесение тонких, труднообнаруживаемых изменений в информационные массивы);
- внедрение программных закладок в другие программы и подпрограммы (вирусный механизм воздействий);
- искажение или уничтожение собственной информации сервера и тем самым нарушение работы сети;
- нарушение целостности (преднамеренное или случайное искажение, разрушение) элементов информационного обеспечения (баз и банков данных), модификация пакетов сообщений и др.

Воздействие этих факторов может иметь катастрофические последствия вплоть до выхода из строя технического, программного и информационного обеспечения систем и, как следствие, привести к нарушениям в процессах обработки инфор-

мации или к отказам функционирования систем в целом [1]. Для нейтрализации последствий воздействия дестабилизирующих факторов может потребоваться реструктуризация системы с перераспределением информационных ресурсов.

В качестве показателя устойчивости функционирования системы в условиях воздействия i -го дестабилизирующего фактора обычно понимается коэффициент устойчивости $K_{yчi}$, определяемый из

выражения $K_{yчi} = 1 - \frac{|K_{тр} - K_{дфи}|}{K_{тр}}$, где $K_{тр}$ – требуемое

значение показателя качества функционирования системы; $K_{дфи}$ – значение показателя качества функционирования системы в условиях воздействия i -го дестабилизирующего фактора [2, 3].

Тогда общий показатель устойчивости системы, выражаемый как коэффициент устойчивости K_y , определяется выражением $K_y = \prod_{i=1}^I K_{yчi}$, где I – количество учитываемых дестабилизирующих факторов в системе.

Кроме того, за показатели качества функционирования системы, характеризующие ее устойчивость, могут быть взяты время выполнения рассматриваемого процесса в системе, коэффициент готовности системы, коэффициент эффективности системы и другие [3].

В настоящее время наиболее эффективными для обеспечения устойчивости функционирования РИС являются следующие подходы:

- совершенствование *процессов хранения и обработки информации* (ПХОИ), в том числе в части обеспечения их устойчивости;
- создание высоконадежных подсистем хранения и обработки данных с учетом их резервирования [4].

Для повышения устойчивости ПХОИ на этапах проектирования и эксплуатации РИС целесообразно выполнять рациональное распределение элементов ПО и информационных массивов по узлам вычислительной сети, а при воздействии дестабилизирующих факторов, в случае структурной деградации сети – их перераспределение [4].

Основу РИС составляет *система хранения и обработки данных* (СХОД), предназначенная для организации надежного, а также отказоустойчивого хранения данных, высокопроизводительного доступа серверов к устройствам хранения и обработки информации. Обеспечение сохранности информации в СХОД производится путем применения специальных мер организации хранения, восстановления (регенерации) информации, специальных устройств резервирования. Качество обеспечения сохранности информации зависит от ее целостности (точности, полноты) и готовности к постоянному использованию [5–7].

Наиболее простые и эффективные способы обеспечения сохранности информации в СХОД –

создание резервных хранилищ данных и реализация мероприятий по оперативному и восстановительному резервированию информации [7].

По типу устройств резервирования резервные хранилища информации в СХОД [5] делятся на хранилища на базе HDD/SSD-дисков (RAID-массивы), на магнитных лентах, на базе оптических дисков.

По количеству уровней резервирования резервные хранилища делятся на одноуровневые и многоуровневые.

При большом количестве разновидностей систем и средств высоконадежного хранения данных, а также их конфигураций стоит задача выбора их состава с учетом решаемых СХОД задач.

Решение задач организации ПХОИ позволяет повысить эффективность функционирования системы за счет снижения издержек, связанных с разрушением информации, включая потери от невозможности решения задач и получения недостоверных выходных результатов, за счет уменьшения времени и затрат на восстановление разрушенной или искаженной информации, рационального использования внешней памяти вычислительных средств для размещения данных РИС и т.д.

Для рациональной организации ПХОИ в РИС в структуре СХОД создают *центры хранения и обработки информации* (ЦХОИ), основной задачей которых является реализация эффективных с точки зрения решаемых в системе задач ПХОИ.

Для обеспечения сохранности информации целесообразно в ЦХОИ создавать подсистемы хранения и обработки данных, основу которых составляют технические средства хранения и обработки информации, интегрируемые в ЦХОИ. В качестве элементов ПО рассматривались программные комплексы, программные модули, осуществляющие решение функциональных задач с использованием информационных массивов в РИС.

Вследствие ограниченности материальных средств, выделяемых на создание РИС, хранение и обработка информации в них осуществляются в ограниченном количестве ЦХОИ, создаваемых в соответствующих узлах сети. В этих центрах аккумулируются мощные вычислительные средства и средства хранения данных, обеспечивающие решение функциональных задач как отдельного сегмента сети, так и системы в целом.

Таким образом, для обеспечения устойчивости функционирования РИС разработан комплекс математических моделей оптимизации

- распределения элементов ПО функциональных задач по узлам сети по критерию минимума объема информации, передаваемой по каналам связи при решении всех функциональных задач;
- распределения информационных ресурсов по ЦХОД по критерию минимума общего значения среднего времени восстановления информации в СХОД;

- состава технических средств СХОД по критерию максимума значения коэффициента готовности системы хранения данных в целом;
- распределения резерва информационных ресурсов по ЦХОД по критерию минимума среднего времени решения всех задач в РИС [8].

Данные задачи относятся к классу задач комбинаторной оптимизации [9], а именно целочисленного дискретного программирования с булевыми переменными [9, 10].

В общем случае математические модели обеспечения устойчивости функционирования РИС могут быть следующими [11].

Определить такие значения x_j ($j=1, 2, \dots, N$), что

$$F(X) = \min \sum_{j=1}^N c_j x_j \quad (1)$$

при ограничениях

$$x_j = \{0, 1\}, j = 1, 2, \dots, N; \quad (2)$$

$$\sum_{j=1}^N a_{ij} x_j \leq b_i, i = 1, 2, \dots, M, \quad (3)$$

где N – число переменных; M – число ограничений.

Для обеспечения эффективности практического применения разработанных математических моделей к методам их решения должны предъявляться определенные требования:

- возможность получения оптимального (квазиоптимального) решения в минимально возможное время;
- возможность управления точностью полученного решения в условиях ограничения на время решения задач;
- невысокая вычислительная сложность;
- возможность распараллеливания процесса получения оптимального (квазиоптимального) решения;
- простота программной реализации вычислительных процедур алгоритма;
- возможность практического применения на вычислительных комплексах с ограниченными характеристиками (невысокой производительности).

Для получения точного решения задачи (1)–(3) существует достаточно большое количество методов дискретной оптимизации [9, 10], наиболее распространенным и наиболее широко используемым из которых является метод ветвей и границ [10].

Основными процедурами, составляющими метод ветвей и границ, являются построение дерева возможных вариантов решения, определение оценки границы решения для каждой вершины дерева и отсечение бесперспективных вершин, которые не могут привести к улучшению рекорда [10].

Общая эффективность метода ветвей и границ при решении задач обеспечения устойчивости функционирования РИС непосредственно зависит от выбора стратегии ветвления переменных и способа оценки границ решения, а также от способа организации компьютерных вычислений [10, 12]. Результаты этого выбора зависят от конкретных условий решения данных задач оптимизации.

Для формирования дерева вариантов в ходе решения задачи наиболее широко используют такие стратегии ветвления переменных [12], как фланговая (левофланговая, правофланговая), фронтальная, локально-избирательная и глобально-поисковая.

Выбор стратегии ветвления оказывает влияние на требуемый для хранения дерева вариантов при решении задачи объем оперативной памяти вычислительного комплекса, а также (за счет особенностей реализации поисковых процедур на дереве решений) на время решения.

Эффективность вычислительных алгоритмов зависит от точности и простоты способа определения верхней или нижней границы возможных решений [10, 12]. Точность определения верхней (нижней) границы целевых решений напрямую влияет на число бесперспективных вариантов, которые отсеиваются в процессе решения. Увеличение точности расчета границ решений связано с ростом вычислительной сложности алгоритмов и, как следствие, – объема вычислений.

В качестве методов оценки границ решения используют точные методы (симплекс-метод и его модификации) и приближенные методы решения двойственной задачи.

Применение теории двойственности при решении задач методом ветвей и границ позволяет использовать для оценки границы решения приближенные алгоритмы [10]. Однако этот способ оценки границы решения эффективен только при простом и достаточно точном алгоритме решения двойственной задачи.

При решении задачи (1)–(3) методом ветвей и границ условие целочисленности переменных (2) можно заменить условием

$$0 \leq x_j \leq 1. \quad (4)$$

Двойственной задачей по отношению к (1), (2), (4) является задача

$$Z = \min \left(\sum_{i=1}^M b_i y_i + \sum_{i=M+1}^{M+N} y_i \right) \quad (5)$$

при ограничениях

$$\sum_{i=1}^M a_{ij} y_i + y_{M+j} \geq c_j, j = 1, 2, \dots, N, \quad (6)$$

$$y_i \geq 0, i = 1, 2, \dots, M+N. \quad (7)$$

В работе [6] предложен приближенный алгоритм оценки границ решения в методе ветвей и границ, основанный на градиентной процедуре решения задачи, двойственной по отношению к исходной. В данном алгоритме использование свойств двойственной задачи и особенностей реализации градиентной процедуры позволяет в процессе решения осуществлять оценку вклада каждой переменной в результат и за счет этого определять порядок ветвления переменных. Для повышения эффективности метода ветвей и границ предложено однократное решение задачи, двойственной по отношению к исходной, с применением данного ал-

горитма для определения порядка ветвления переменных [10, 13].

Экспериментальная оценка эффективности методов решения задач обеспечения устойчивости функционирования РИС проводилась на вычислительной системе со следующими характеристиками: ОС Windows 7 Professional x64 Service Pack 1; процессор Intel Core i5-3450 с тактовой частотой 3.10 Ghz; объем оперативной памяти 8 Gb.

При экспериментальной проверке методов и алгоритмов решения задач обеспечения устойчивости функционирования РИС методом ветвей и границ ставились следующие задачи:

- оценка эффективности стратегий ветвления переменных в методе ветвей и границ;
- оценка эффективности применения предложенного способа сокращения времени решения задач методом ветвей и границ на основе применения теории двойственности;
- определение параметров *генетического алгоритма* (ГА), обеспечивающих наилучшее качество получаемого решения;
- определение влияния исходных данных на эффективность методов;
- определение области эффективного применения методов решения формализованных математических моделей.

Исходные данные для экспериментальной проверки эффективности методов решения разработанных математических моделей формировались следующим образом.

Коэффициенты при переменных в целевой функции и ограничениях генерировались как независимые друг от друга целые случайные числа, равномерно распределенные на интервале [0, 100]. Правые части ограничений вычислялись по зависимости

$$b_i = \frac{Q}{100} \sum_{j=1}^N a_{ij}.$$

Коэффициент Q определял жесткость ограничений и выбирался из интервала [30, 90] с шагом 20 %. Для каждой размерности решалось по 10 задач.

В качестве характеристик метода ветвей и границ [14, 15] использовались следующие величины: время решения задачи (t_{\min} – минимальное, t_{\max} – максимальное, $t_{\text{ср}}$ – среднее); число просмотренных вершин (V_{\min} – минимальное, V_{\max} – максимальное, $V_{\text{ср}}$ – среднее); общее время определения границ решения выбранным методом (T_{\min} – минимальное, T_{\max} – максимальное, $T_{\text{ср}}$ – среднее).

Для оценки эффективности метода ветвей и границ при решении разработанных задач использовался разработанный программный комплекс [16].

Результаты экспериментальной оценки эффективности применения метода ветвей и границ для решения задач различных размерностей приведены на рисунках 1–4.

Из представленных данных видно, что наименьшее время решения задач обеспечивают глобально-

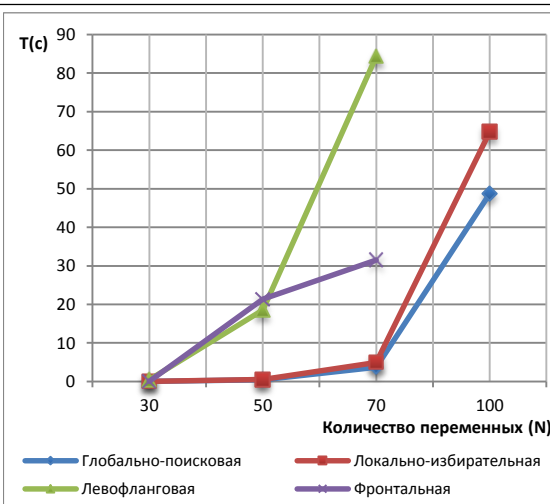


Рис. 1. Зависимость времени решения от размерности задач при количестве ограничений $M = 20$ и жесткости $Q = 50$ %

Fig. 1. Dependence of solution time on a problem dimension with the number of restrictions $M = 20$, and the rigidity $Q = 50$ %

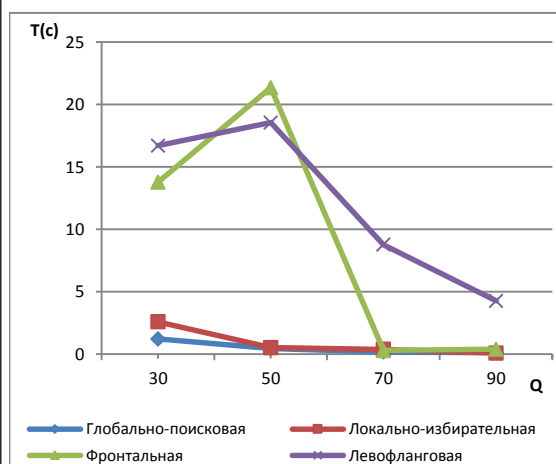


Рис. 2. Зависимость времени решения задач (50×20) от жесткости ограничений Q

Fig. 2. Dependence of time for solving problems (50×20) on stiffness of constraints Q

поисковая и локально-избирательная стратегии ветвления. Наименее эффективной является фланговая стратегия ветвления. Следует отметить нестабильность поведения фронтальной стратегии ветвления.

При некоторых условиях использование фронтальной стратегии приводит к тому, что процесс решения задачи сводится к полному перебору вариантов решения. Поведение фронтальной стратегии ветвления не зависит от исходных данных решаемых задач и выбора метода оценки границ решения.

Результаты оценки влияния жесткости ограничений на эффективность стратегий ветвления пред-

Таблица 1

**Оценка влияния предварительного определения
порядка ветвления переменных на эффективность метода ветвей и границ ($Q = 0,5$)**

Table 1

**Impact assessment of the preliminary determination of the branching order of variables on the efficiency
of the branch and bound method ($Q = 0,5$)**

Размерность задачи	Среднее время решения t (сек.)			
	Без определения порядка ветвления		С определением порядка ветвления	
	Симплекс-метод	Приближенный метод	Симплекс-метод	Приближенный метод
30×10	0,13	0,24	0,04	0,06
50×10	0,54	60,16	0,39	5,34
60×10	3,95	76,69	1,73	7,04
70×10	16,87	284	2,73	8,06
100×10	367,14	-	26,31	4002,68

ставлены на рисунках 2 и 3. На них видно, что со снижением жесткости ограничений число просмотренных вершин и, соответственно, время решения задач уменьшаются. Это влияние особенно сильно проявляется с увеличением размерности задачи (числа переменных). Так, на размерности 100 переменных и 30 ограничений снижение жесткости ограничений с $Q = 50\%$ до $Q = 90\%$ сокращает время решения задач в 29 раз. Это объясняется снижением количества перспективных вершин среди не ответвленных на дереве решений. Таким образом, повышение жесткости ограничений приводит к значительному увеличению числа просматриваемых методом вариантов решения, что ставит под сомнение целесообразность использования в этих случаях таких стратегий ветвления переменных, как фланговая, фронтальная и их модификации.

Анализ результатов расчетов, представленных на рисунке 3, показывает, что в зависимости от выбранной стратегии ветвления 27–89 % от времени решения задачи составляет совокупное время определения границ решения симплекс-методом.

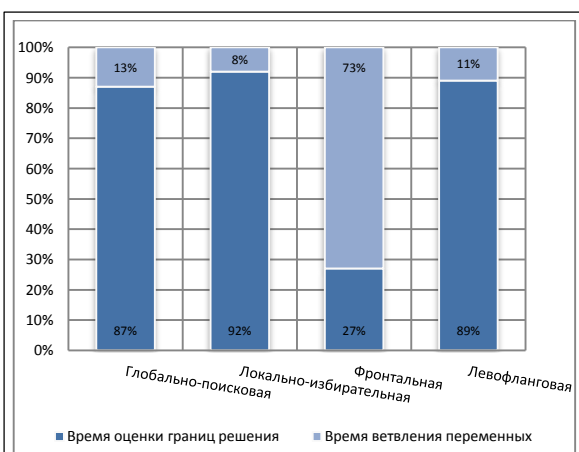


Рис. 3. Доля времени ветвления переменных относительно общего времени решения задачи (70×10)

Fig. 3. The variable branch-time share compared with total problem solving time (70×10)

При оценке влияния применения алгоритма определения порядка ветвления переменных на эффективность метода ветвей и границ в качестве методов оценки границ решения использовались симплекс-метод и приближенный метод решения двойственной задачи.

Результаты экспериментальной проверки эффективности влияния предварительного определения порядка ветвления переменных приведены в таблице 1 и на рисунке 4 [13–15]. В качестве методов оценки границ решения использовались симплекс-метод и рассмотренный выше приближенный метод решения двойственной задачи на основе применения градиентной процедуры, в качестве стратегии ветвления переменных – глобально-поисковая стратегия.

Из таблицы 1 и рисунка 4 видно, что предварительное определение порядка ветвления переменных при решении задач различных размерностей позволяет снизить время решения задач в 3–14 раз. Результаты решения стабильны независимо от способа оценки верхней границы. Это говорит об эф-

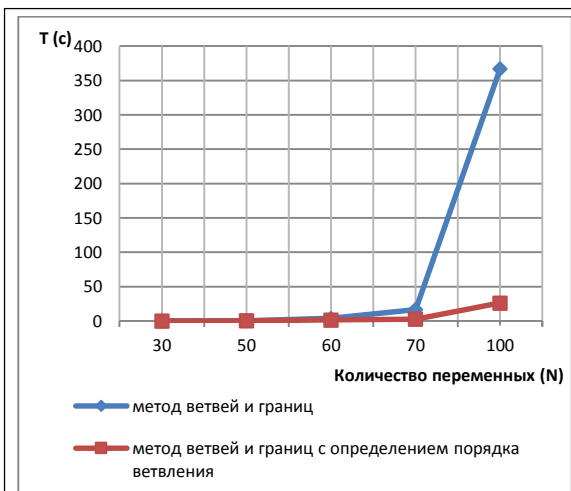


Рис. 4. Зависимость времени решения задач ($m=10$) от количества переменных в целевой функции

Fig. 4. The problem solving time ($m = 10$) depending on the number of variables in the objective function

фактивности предложенного алгоритма и целесообразности его использования при решении практических задач методом ветвей и границ.

Таким образом, экспериментальная проверка эффективности метода ветвей и границ показала, что наиболее эффективными при решении задач обеспечения устойчивости РИС являются глобально-поисковая и локально-избирательная стратегии ветвления переменных. Увеличение жесткости ограничений приводит к значительному росту времени решения задач методом ветвей и границ независимо от используемых стратегий ветвления и методов оценки границ решения. Применение предложенного способа ранжирования переменных по уровню влияния на результат решения задачи для организации процесса ветвления переменных совместно с точным методом оценки границ решения обеспечивает существенное (от 3 до 14 раз) повышение производительности метода ветвей и границ.

Достоинством метода ветвей и границ является точность получаемого решения за ограниченное время, недостатком – экспоненциальная зависимость времени решения от размерности решаемых задач [15].

Для практического применения формализованных задач обеспечения устойчивости функционирования РИС решения задачи в условиях жестких ограничений на время решения предложено и экспериментально проверено использование ГА как варианта стохастического эволюционного поиска решения. Основу ГА составляет хромосома (особь) [17, 18].

Применительно к данной задаче хромосома (особь) состоит из набора N генов, каждый из которых представляет собой значение соответствующей переменной решаемой задачи и принимает значение 0 или 1.

В ГА над хромосомами выполняются следующие действия: определение значения функции приспособленности, скрещивание (кроссинговер), мутация, отбор особей для скрещивания.

Значение функции приспособленности используется для оценки качества варианта решения, соответствующего значениям генов хромосомы.

Функция приспособленности рассчитывается как значение целевой функции.

Если геном хромосомы соответствует недопустимому значению переменных (не выполняются ограничения), функция приспособленности возвращает значение < 0 (например $-100\,000$), в противном случае – положительное число, значение W .

Для генерации новых вариантов решения (особей) в ГА используют механизмы отбора особей, скрещивания и мутации хромосом.

Для работы ГА [17–19] обычно используют следующие популяции:

- исходная популяция; служит для хранения особей, участвующих в текущем шаге ГА; на

начальном шаге ГА исходная популяция обычно генерируется случайным образом;

- родительская популяция (родительский пул); в родительскую популяцию отбираются все особи исходной популяции, значения функции приспособленности которых не меньше среднего значения функции приспособленности для исходной популяции в целом;

- элитная популяция; служит для сохранения лучших особей родительской (исходной) популяции;

- дочерняя популяция; формируется путем многократного выполнения операций отбора особей, кроссинговера, мутации; на основе дочерней популяции формируется исходная популяция для следующего шага ГА.

Для упрощения процедур ГА предлагается использовать исходную популяцию постоянного размера. В случае, если на какой-либо итерации вновь сформированная популяция, которая на следующей итерации будет использоваться как исходная, имеет размер, менее заданного, предлагается дополнять ее до требуемого размера случайно сгенерированными особями.

Особи для скрещивания выбираются из родительской популяции. Первая хромосома выбирается случайным образом.

Выбор второй особи для скрещивания возможен по следующим вариантам:

- случайный выбор; вторая особь выбирается случайно из родительской популяции; данный вариант наиболее прост для реализации, однако вследствие своей природы не может гарантировать качество получаемого решения;

- родственное скрещивание на основе анализа расстояния между особями; в качестве расстояния между хромосомами для решаемой задачи целесообразно использовать расстояние Хэмминга, определяемое для двух особей как $S = \sum_{i=1}^N |g_i^1 - g_i^2|$, где g_i^1 и g_i^2 – значения i -х генов 1-й и 2-й особей.

В соответствии с этим различают близкородственное (инбридинг) и дальнеродственное (аутбридинг) скрещивание.

Аутбридинг – выбирается хромосома с максимальным значением расстояния Хэмминга до первой выбранной: $S \rightarrow \max$. Данный вид отбора обеспечивает снижение скорости сходимости алгоритма к локальному экстремуму и увеличивает вероятность достижения глобального экстремума.

Инбридинг – выбирается хромосома с минимальным расстоянием от выбранной первой: $S \rightarrow \min$. Данный вид отбора обеспечивает максимально быструю сходимость алгоритма к локальному экстремуму.

Для обеспечения сохранения лучших найденных особей содержимое элитной популяции автоматически включается в дочернюю популяцию на текущем шаге алгоритма.

Размер элитной популяции L^{el} определяется как $L^{el} = k^{el} L^{sr}$, где k^{el} – коэффициент элитизма; L^{sr} – размер исходной популяции.

Операция скрещивания (кроссинговер) выполняется над парой хромосом, являющихся родительскими. Результат операции – две хромосомы-потомка.

Исследования показали, что одним из наиболее эффективных с точки зрения конечного результата является универсальный кроссинговер, который выполняется следующим образом.

Генерируется шаблон двоичных чисел длиной N . Шаблон указывает, какие гены должны наследоваться от первого родителя (остальные гены берутся от второго родителя).

Для первого потомка: если в j -й позиции шаблона 1, то j -й ген для данного потомка берется от первого родителя, если 0 – от второго. Для второго потомка: если в j -й позиции шаблона 1, то j -й ген для данного потомка берется от второго родителя, а если 0 – от первого.

Кроме того, потенциально хороший результат может показать триадный кроссинговер с формированием маски кроссинговера из лучшей особи элитной популяции. Он аналогичен универсальному, за исключением того, что маска кроссинговера формируется не случайно, а соответствует лучшей особи в элитной популяции.

В качестве претендента на использование возможно применение триадного кроссинговера на основе анализа схемы [20] в следующем виде.

Пусть g^1 , g^2 и g^3 – родительские хромосомы; f_{g^1} , f_{g^2} , f_{g^3} – значения функции приспособленности родительских особей. g^1 и g^2 выбираются в соответствии с текущей схемой отбора особей, g^3 – случайным образом из особей родительской популяции.

Так как гены принимают только значения из набора $\{0, 1\}$, обозначим через $P_0(j)$ и $P_1(j)$ частоты появления значений 0 и 1 для j -го гена в родительской популяции соответственно. Тогда для всех j , $j=1, 2, \dots, N$, определим значения генов дочерних особей \bar{g}_j^1 и \bar{g}_j^2 по следующим правилам.

Если $g_j^1 = g_j^2$, то $\bar{g}_j^1 = g_j^1$.

Если $g_j^1 \neq g_j^2$, то

при $g_j^1 = 0$, если $f_{g^1} P_0(j) \geq f_{g^2} P_1(j)$, $\bar{g}_j^1 = g_j^1$ и $\bar{g}_j^2 = g_j^2$ в противном случае;

при $g_j^1 = 1$, если $f_{g^1} P_1(j) \geq f_{g^2} P_0(j)$, $\bar{g}_j^1 = g_j^1$ и $\bar{g}_j^2 = g_j^2$ в противном случае.

Если $f_{g^1} > f_{g^2}$, то $g^4 = g^1$, $f_{g^4} = f_{g^1}$ и $g^4 = g^2$, $f_{g^4} = f_{g^2}$ в противном случае.

Если $g_j^4 = g_j^3$, то $\bar{g}_j^2 = g_j^4$.

Если $g_j^4 \neq g_j^3$, то

при $g_j^4 = 0$, если $f_{g^4} P_0(j) \geq f_{g^3} P_1(j)$, $\bar{g}_j^2 = g_j^4$ и $\bar{g}_j^1 = g_j^3$ в противном случае;

при $g_j^4 = 1$, если $f_{g^4} P_1(j) \geq f_{g^3} P_0(j)$, $\bar{g}_j^2 = g_j^4$ и $\bar{g}_j^1 = g_j^3$ в противном случае.

Нетрудно заметить, что при $P_1(j) = 1$ или $P_0(j) = 1$ все особи в популяции имеют одинаковое значение j -го гена и это значение гарантированно перейдет по наследству потомкам.

Операция мутации выполняется над выбранной хромосомой. Данный оператор необходим для «выбывания» популяции из локального экстремума и препятствует преждевременной сходимости. Это достигается за счет того, что изменяется случайно выбранный ген g_i ($i = 1, 2, \dots, N$) в хромосоме. Значение выбранного гена (переменной исходной задачи в данном варианте решения) инвертируется.

Введем обозначения: N^{gen} – число поколений, формируемых островом в одной итерации ГА; P_c – вероятность выполнения скрещивания хромосом; P_m – вероятность мутации хромосомы.

Простой ГА состоит из следующих шагов.

1. Начальная инициализация исходной популяции случайно сформированными хромосомами.

2. $k = 0$.

3. Определение среднего значения функции принадлежности хромосом исходной популяции.

4. Формирование родительской популяции из особей исходной популяции, значение функции принадлежности которых не меньше среднего значения функции принадлежности в исходной популяции. Определение длины родительской популяции L^{pr} .

5. Формирование элитной популяции из L^{el} лучших особей исходной популяции.

6. Включение в дочернюю популяцию хромосом, входящих в элитную популяцию.

7. $i = 0$.

8. Случайный выбор хромосомы g_1 из родительской популяции.

9. В соответствии с заданным способом выбора второй хромосомы для скрещивания выбор из родительской популяции хромосомы g_2 .

10. Формирование случайного числа $rnd \in [0, 1]$. Если $rnd \leq P_c$, выполнение процедуры скрещивания хромосом g_1 и g_2 с получением новых хромосом \bar{g}_1 и \bar{g}_2 , для которых выполняется расчет значения функции принадлежности. Если $rnd > P_c$, присвоение \bar{g}_1 значения g_1 и \bar{g}_2 значения g_2 .

11. Формирование случайных чисел из интервала $[0, 1]$ rnd_1 и rnd_2 . Если $rnd_1 \leq P_m$, выполнение операции мутации с хромосомой \bar{g}_1 . Если $rnd_2 \leq P_m$, выполнение операции мутации с хромосомой \bar{g}_2 .

12. Включение с учетом результатов проверки на совпадение хромосом и допустимость значения функции принадлежности хромосом \bar{g}_1 и \bar{g}_2 в дочернюю популяцию.

13. $i = i + 1$.
14. Если $i < L^p/2$, переход к шагу 8, в противном случае – к шагу 15.
15. Дополнение дочерней популяции хромосомами из родительской популяции, не входящими в состав дочерней. Определение размера дочерней популяции L^{ch} .
16. Если $L^{ch} < L^{sr}$, дополнение дочерней популяции случайно сформированными хромосомами в количестве $L^{sr} - L^{ch}$.
17. Очистка исходной, родительской и элитной популяций. Перенос хромосом из дочерней популяции в исходную. Очистка дочерней популяции.
18. $k = k + 1$.
19. Если $k < N^{gen}$, переход к шагу 3, иначе – к шагу 20.
20. Формирование элитной популяции.
21. Конец. Результат – лучшая особь в элитной популяции.

При решении задач обеспечения устойчивости функционирования РИС рассматривались две схемы генерации особей для формирования исходных популяций:

- случайная генерация варианта решения (особи) с последующей проверкой на выполнимость ограничений;
- последовательная генерация варианта решения (особи) с поэтапной проверкой выполнимости ограничений.

Обе схемы основаны на случайной генерации значений переменных при решении задач обеспечения устойчивости функционирования РИС.

Первая схема потенциально имеет больше шансов сформировать оптимальное решение уже в начальной популяции, однако требует существенного отсева вариантов, не удовлетворяющих ограничениям.

В соответствии со второй схемой, если на последующем этапе полученные значения переменных в варианте решения задач обеспечения устойчивости функционирования РИС не удовлетворяют ограничениям, за вариант решения берется особь, полученная на предыдущем этапе. Данная схема формирует исходную популяцию за количество шагов, незначительно превышающее размер популяции.

Экспериментальная проверка показала, что реализация первой схемы занимает до 30 % от общего времени решения задачи. Причем с ростом размерности задачи, количества ограничений, жесткости ограничений эта доля времени существенно возрастает.

Применение второй схемы позволяет сократить долю времени на формирование исходной популяции до 1–2 % от общего времени решения. Причем при ужесточении или росте числа ограничений данная величина растет незначительно.

Таким образом, при решении практических задач целесообразно применять схему последова-

тельного формирования вариантов решения (особей) при создании исходной популяции.

Определено, что наилучшие показатели при решении задач обеспечения устойчивости функционирования РИС простой ГА имеет при $P_c = 0,8$, $P_m = 0,1$, $k_{el} = 0,1$.

Выбор схемы репродукции оказывает существенное влияние на качество получаемого результата. Однако эффективность схемы репродукции существенно зависит от результатов начальной инициализации исходной популяции. Неверный выбор схемы репродукции обычно приводит к существенному ухудшению качества получаемого решения. В связи с тем, что инициализация исходной популяции осуществляется случайным образом, заранее осуществить эффективный выбор схемы репродукции не представляется возможным. Поэтому предлагается использовать адаптивную схему репродукции. Для этого в каждую хромосому необходимо включить числовые метки используемых для ее получения схем отбора и варианта кроссинговера.

На каждом шаге алгоритма (при формировании нового поколения особей) необходимо определить количество особей в родительской популяции, сформированных с использованием каждой схемы отбора и каждого вида кроссинговера, и рассчитать частоту (вероятность) использования схем отбора и вариантов кроссинговера.

При начальной инициализации исходной популяции для всех хромосом данным меткам присваивается значение -1 , а частоты применения схем отбора и вариантов кроссинговера принимают равновероятными. При этом схемы отбора особей и варианты кроссинговера используются случайным образом в соответствии с полученными частотами (вероятностями) их применения.

Для реализации адаптивной схемы репродукции рассматривались схемы отбора особей: аутбридинг, инбридинг, случайный отбор, а также варианты кроссинговера: одноточечный, двухточечный, универсальный, триадный (с формированием маски кроссинговера из лучшей особи элитной популяции), триадный с учетом схемы.

В связи с незначительным влиянием мутации особей на качество получаемого решения использовался только один вариант мутации – одноточечная мутация.

Экспериментальная проверка эффективности ГА осуществлялась с применением разработанной программы [21]. Результаты экспериментальной проверки представлены в таблицах 2, 3 и на рисунке 5.

Из таблиц 2 и 3 видно, что качество получаемого решения при использовании простого ГА вследствие сходимости к случайному локальному экстремуму может существенно варьироваться в зависимости от результата начальной инициализации исходной популяции [22].

Таблица 2
Время формирования 1-го поколения ГА (сек.)
Table 2
Time of formation of the 1st generation of GA (sec.)

Размерность задачи	Схема отбора особей			
	Аутбридинг	Инбридинг	Случайный	Адаптивный
70×4	0,00207	0,00205	0,00207	0,00208
100×4	0,00376	0,00376	0,00371	0,00375
150×4	0,008745	0,008761	0,008381	0,008636

Экспериментальная проверка показала, что простой ГА с высокой долей вероятности позволяет получить оптимальное решение для задач малой (до 70 переменных) размерности. В случае получения в качестве решения локального экстремума отклонение от оптимума при том же времени решения составляет до 1,5 %.

На рисунке 5 представлена зависимость времени формирования одного поколения особей в ГА от размерности (числа переменных) решаемой задачи.

Из таблицы 2 и рисунка 5 видно, что с ростом числа переменных решаемой задачи время формирования одного поколения ГА растет экспоненциально. Данное обстоятельство позволяет судить о том, что время решения задач ГА будет нелинейно расти с ростом размерности задач. Степень нелинейности зависит от выбранных параметров алгоритма. Кроме того, с ростом размера популяции время решения задачи растет в квадратичной зависимости. Качество получаемого решения при использовании простого ГА вследствие сходности к случайному локальному экстремуму может существенно варьироваться в зависимости от результата начальной инициализации исходной популяции [17, 22].

Кроме того, из таблицы 3 видно, что с ростом размерности задачи выше 100 переменных область поиска решения столь существенно расширяется,

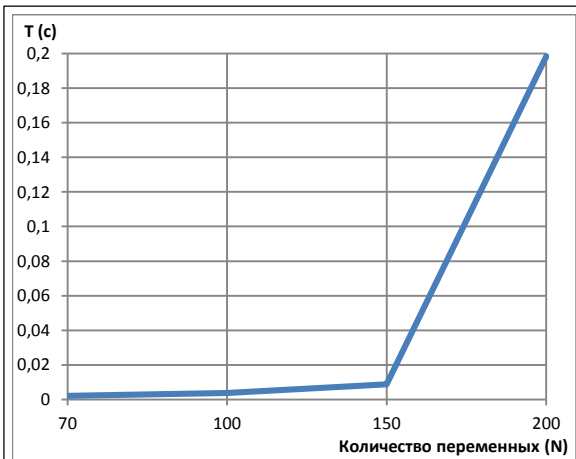


Рис. 5. Зависимость времени формирования 1-го поколения от размерности задачи (случайный отбор)

Fig. 5. Dependence of the 1st generation formation time on a problem dimension (random selection)

что вероятность получения оптимального решения простым ГА значительно снижается независимо от выбора параметров алгоритма.

Наилучшие схемы репродукции – случайный отбор и адаптивная (обеспечивают наискорейшее достижение оптимального решения при минимальном количестве итераций). При этом эффективность адаптивной схемы растет по сравнению со случайной с ростом размерности задачи. Это объясняется несколько большим разнообразием особей в исходной популяции, получаемым при использовании адаптивной схемы. Данная схема показывает более стабильный по качеству получаемого решения результат. При этом время решения незначительно выше, чем при использовании случайной схемы.

В связи с тем, что с ростом размерности решаемых задач простой ГА в большинстве случаев получает в качестве решения локальный экстремум, значение и близость которого к оптимальному существенно зависят от результатов начальной инициализации.

Таблица 3
Результаты экспериментальной проверки эффективности простого ГА для решения задачи обеспечения устойчивости функционирования РИС

Table 3
Results of experimental verification of the simple GA effectiveness to solve the task of ensuring the sustainability of distributed information systems

Размерность задачи	Аутбридинг		Инбридинг		Случайный		Адаптивный	
	Вероятность нахождения опт. рещ. (%)	Среднее время решения (сек.)	Вероятность нахождения опт. рещ. (%)	Среднее время решения (сек.)	Вероятность нахождения опт. рещ. (%)	Среднее время решения (сек.)	Вероятность нахождения опт. рещ. (%)	Среднее время решения (сек.)
70×4	0,2	1,982	0,4	1,971	1,0	1,983	0,8	2,001
100×4	0	12,032	0	12,02	0,2	11,861	0,2	11,983
150×4	0	69,96	0,1	70,088	0	67,048	0	69,09

циализации исходной популяции, для получения более стабильного результата предлагается использовать *островной ГА* (ОГА), являющийся модификацией простого ГА [23].

Сущность ОГА заключается в следующем. Одновременно и независимо друг от друга (параллельно) работают несколько простых ГА со своими популяциями. Каждый из них вследствие случайной начальной инициализации исходной популяции сходится к своему локальному экстремуму (острову). С определенной периодичностью (через определенное число поколений) по определенной схеме ГА обмениваются заданным числом лучших особей (осуществляется миграция особей), после чего процесс решения продолжается. Совокупность шагов алгоритма, завершающаяся обменом лучшими особями между островами (миграцией особей), составляет одну итерацию ОГА. Подобная схема позволяет обеспечить выход простых ГА в каждом из островов из области локального экстремума, обеспечить большее разнообразие особей в популяциях и с большей вероятностью получить лучшее решение (в том числе достичь глобального экстремума). В качестве схемы миграции особей чаще всего используются миграция с топологией полной сети, миграция с топологией кольца и их модификации [22, 23].

Введем обозначения: N^{Lnd} – число островов; N^{Step} – число итераций ОГА.

ОГА с миграцией особей с топологией кольца состоит из следующих шагов.

1. Для каждого k -го острова ($k = 1, 2, \dots, N^{Lnd}$) выполнить инициализацию исходной популяции путем заполнения ее особями, сгенерированными случайным образом (в исходную популяцию не включаются нежизнеспособные особи, повторное включение уже имеющейся особи не допускается) и формирование элитной популяции объемом L^{el} особей.

2. $i = 0$.

3. Для каждого острова ($k = 1, 2, \dots, N^{Lnd}$) осуществляется последовательная эволюция поколений в количестве N^{gen} (выполнение простого ГА с заданной схемой репродукции).

4. Если $N^{Lnd} > 1$, для каждого k -го острова ($k = 1, 2, \dots, N^{Lnd} - 1$) осуществляется перенос m ($m = L^{el}/2$) лучших особей из элитной популяции в исходную популяцию j -го ($j = k + 1$) острова путем замещения худших особей. Для случая $k = N^{Lnd}$ осуществляется перенос лучших особей в 1-й остров.

5. $i = i + 1$.

6. Если $i < N^{Step}$, перейти к п. 3. В противном случае к п. 7.

7. Сформировать результирующую популяцию из особей, составляющих элитные популяции каждого из островов.

8. Из результирующей популяции выбрать лучшую особь – результат решения.

9. Конец.

Помимо адаптивной схемы репродукции, возможно использование ее упрощенного варианта – смешанной или гибридной схемы, при которой вариант селекции и кроссинговера для каждого острова выбирается при инициализации случайно и однократно.

Результаты экспериментальной проверки эффективности ОГА для решения задач дискретного линейного целочисленного программирования с булевыми переменными представлены в таблицах 4, 5 и на рисунках 6–8.

В таблице 4 показаны экспериментально полученные параметры ОГА при решении задач обеспечения устойчивости функционирования РИС различной размерности, обеспечивающие максимально возможные шансы получения оптимального решения [22]. Из таблицы видно, что для получения решения максимального качества в ОГА количество островов должно быть не меньше $0,025N$, количество особей в популяции – $0,6N$, количество поколений в итерации должно быть в интервале от $0,9N$ до $1,2N$, число итераций – не меньше $0,05N$.

В таблице 5 представлены результаты решения задач различной размерности ОГА с применением параметров алгоритма, приведенных в таблице 2. Из таблицы 5 видно, что ОГА с достаточно высокой степенью вероятности (по сравнению с простым ГА) получает оптимальное решение задач, в том числе большой размерности (число переменных более 150). В случае получения локального экстремума отклонение от оптимального решения не превышало 0,2 %.

На рисунке 6 показана зависимость значения целевой функции решаемой задачи (размерность 150×4) от числа итераций. ОГА осуществляет последовательное приближение к глобальному экстремуму. При этом с 4-й итерации скорость прироста качества получаемого решения существенно снижается. Это свидетельствует о том, что в случае

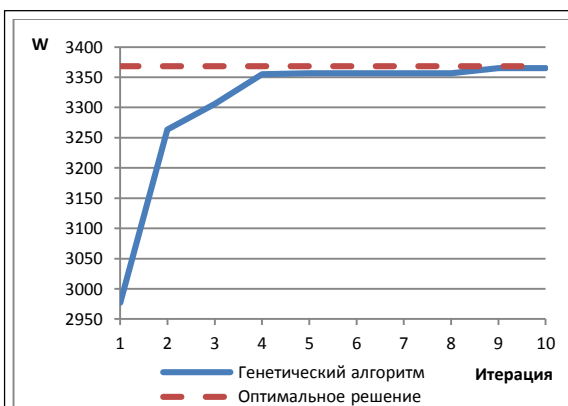


Рис. 6. Зависимость значения целевой функции от итерации (задача 150×4)

Fig. 6. Dependence of the objective function on the iteration (task 150×4)

Таблица 4

Параметры и характеристики ОГА для решения задач дискретного целочисленного программирования с булевыми переменными

Table 4

Parameters and characteristics of an island genetic algorithm for solving the problems of discrete integer programming with boolean variables

Параметры и характеристики	Количество переменных N				Значение параметра
	70	100	150	200	
Количество островов	4	4	5	8	$0,025N$
Количество итераций	6	4	8	10	$0,05N$
Количество особей в популяции	50	60	80	120	$0,6N$
Длина периода миграции особей (поколений)	-	120	160	180	$0,9N \div 1,2N$
Поколений в миграции	40	200	200	200	-
Среднее время решения (сек.)	2,0	12,1	68,8	396,9	-
Среднее время решения задач методом ветвей и границ	0,2	13,99	74,05	1279,58	-

Таблица 5

Результаты экспериментальной проверки эффективности ОГА для решения задачи обеспечения устойчивости функционирования РИС

Table 5

Results of experimental verification of the effectiveness of the island GA for solving the problem of ensuring the sustainability of distributed information systems

Размерность задачи	Аутбридинг		Инбридинг		Случайный		Адаптивный	
	Вероятность нахождения оптималь- ного реше- ния (%)	Среднее время реше- ния (сек.)	Вероятность нахождения оптималь- ного реше- ния (%)	Среднее время реше- ния (сек.)	Вероятность нахождения оптималь- ного реше- ния (%)	Среднее время реше- ния (сек.)	Вероятность нахождения оптималь- ного реше- ния (%)	Среднее время реше- ния (сек.)
70×4	0,70	2,007	0,50	2,00	0,90	2,00	0,90	2,031
100×4	0,00	12,19	0,30	12,18	0,90	11,91	0,75	11,98
150×4	0,10	68,82	0,10	68,79	0,70	66,38	0,70	70,13

жесткого лимита времени, отводимого на решение задачи, когда можно удовлетвориться рациональным (квазиоптимальным) решением, достаточно использования 4-5 итераций алгоритма для данной размерности задач, что позволит сократить время поиска решения не менее чем на 40–60 %.

Время решения задачи ОГА линейно растет с увеличением числа островов.

На рисунках 7 и 8 представлены результаты работы адаптивной схемы в ОГА при решении задачи размерности 100×4.

К 5-6-й итерациям процедура адаптации сходится к выбору одной из схем отбора особей или их устойчивой комбинации. Аналогичная ситуация наблюдается в адаптации выбора кроссинговеров к 6–8-й итерациям алгоритма. Это может свидетель-

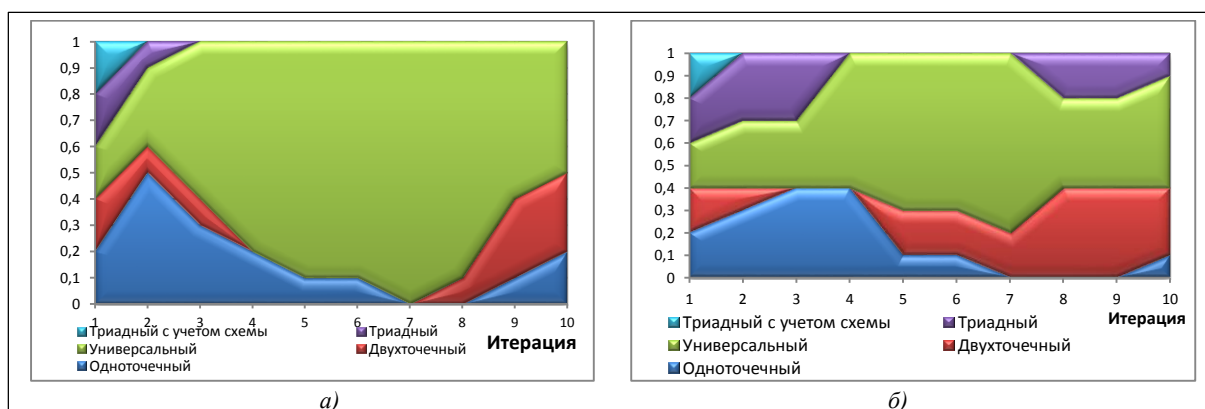
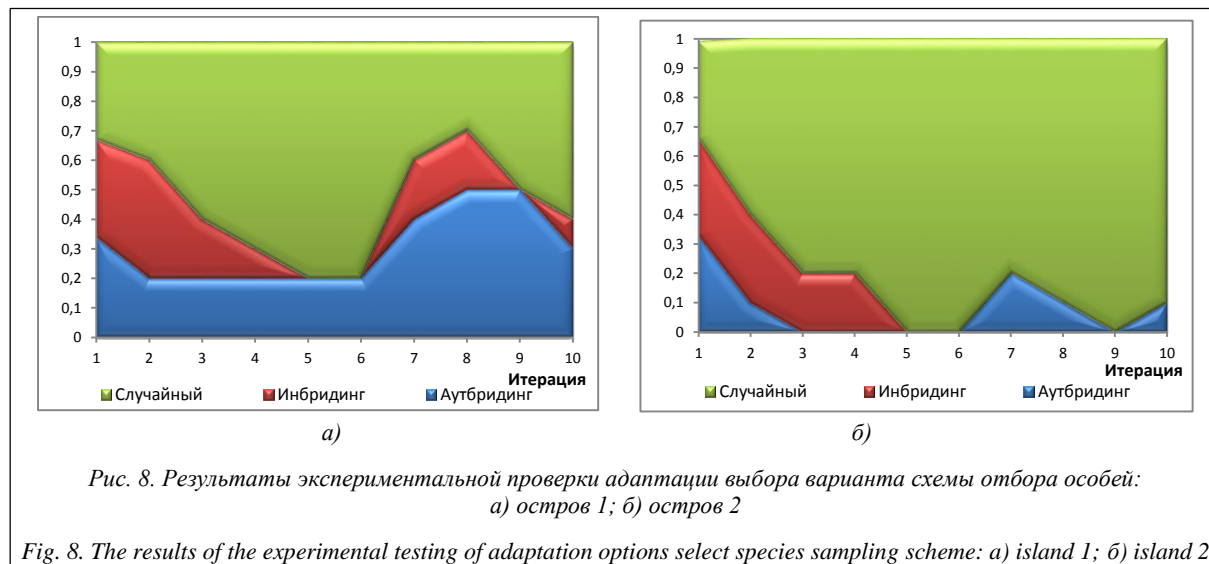


Рис. 7. Результаты экспериментальной проверки адаптации выбора варианта кроссинговера:

а) остров 1; б) остров 2

Fig. 7. The results of experimental verification of selection crossover adaptation options: а) island 1; б) island 2



ствовать о том, что ОГА к 7-й итерации осуществляет достаточное и устойчивое сближение к локальному экстремуму в каждом из островов. При этом миграция особей между островами до схождения к локальному экстремуму не оказывает кардинального влияния на работу схемы адаптации и лишь улучшает состав популяции, обеспечивая снижение скорости схождения к ближайшему локальному экстремуму. После 5–7-й итераций (когда острова в значительной степени приблизились к локальным экстремумам) обмен лучшими особями приводит к кратковременному выходу схемы адаптации из равновесного состояния.

Применение триадных кроссингверов с учетом схемы и с использованием в качестве маски кроссингвера лучшей особи не дают требуемого качества решения; и при использовании адаптивной схемы репродукции, и в большинстве случаев к 3–4-й итерации ОГА отсекаются от применения. При этом количество лучших особей, полученных

с применением универсального кроссингвера, в популяциях во всех островах ОГА составляет порядка 46 %.

На рисунках 9–11 представлена сравнительная оценка влияния исходных данных на время решения задачи ОГА и методом ветвей и границ. Из рисунка 9 видно, что для больших размерностей (более 150 переменных) время решения задачи ОГА растет не так быстро, как при методе ветвей и границ, в том числе с определением порядка ветвления переменных, что говорит об эффективности и перспективности применения ОГА для получения точного решения задач большой размерности.

Из рисунка 10 видно, что с ростом числа ограничений в решаемой задаче время решения ОГА возрастает по линейному закону и носит менее выраженный характер по сравнению с методом ветвей и границ. Для размерности задач с количеством ограничений меньше 4 время решения задачи методом ветвей и границ с определением порядка вет-

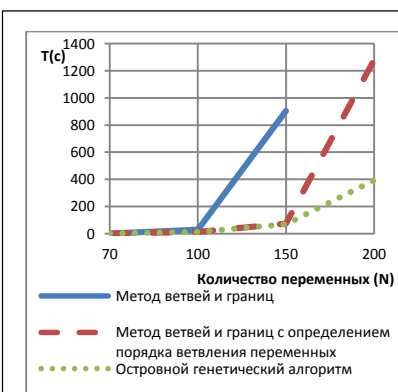


Рис. 9. Зависимость времени решения задач от числа переменных при количестве ограничений $M = 4$

Fig. 9. The time for solving a number of variables when the number of restrictions $M = 4$

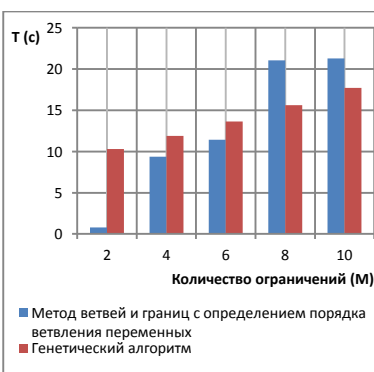


Рис. 10. Влияние числа ограничений на время решения задачи

Fig. 10. The effect of the number of restrictions on the time for solving the problem

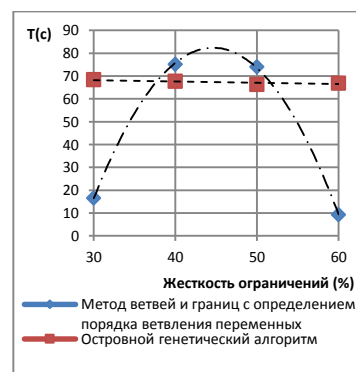


Рис. 11. Зависимость времени решения задачи 150×4 от жесткости ограничений

Fig. 11. Dependence of 150×4 problem solving time on constraint rigidity

Таблица 6

Результаты оценки влияния точности решения задач на время решения

Table 6

Results of assessing the impact of the accuracy of solving problems at the solution time

Размерность задачи	Время получения оптимального решения (сек.)	Оптимальное решение	СКО	Среднее значение целевой функции	Среднее время решения (сек.)	Относительная погрешность решения (%)
100×4	12	2292,61	1,87	2291,57	10,132	0,0454
			3,08	2290,96	7,08	0,072
			3,3	2288,96	4,864	0,1593
			4,45	2277,34	2,218	0,6661
			11,39	2267,02	1,118	1,116
			11,79	2270,52	1,654	0,9637
150×4	69,5	3368,20	1,31	3367,54	37,428	0,0195
			3,85	3365,56	54,524	0,0783
			3,03	3357,33	11,806	0,3226
			5,67	2933,31	3,256	12,9117
			9,43	2917,90	1,642	13,369

ления переменных меньше, чем ГА, это объясняется меньшей вычислительной сложностью данного метода для задач малой размерности (с малым числом ограничений).

На рисунке 11 представлена оценка влияния жесткости ограничений в решаемой задаче на время решения задач. Из рисунка видно, что изменение жесткости ограничений не оказывает существенного влияния на время решения задачи ОГА. При этом время решения задачи ОГА оказывается ниже (до 20 %) времени решения той же задачи методом ветвей и границ с определением порядка ветвления переменных при значении жесткости ограничений в интервале 40–50 %.

В таблице 6 и на рисунке 12 представлены результаты оценки зависимости влияния точности получаемого решения на общее время решения задачи ОГА. Изменение точности получаемого решения осуществлялось путем варьирования значений параметров ОГА, представленных в таблице 2. Для каждого варианта параметров ОГА задача каждой размерности решалась по 10 раз.

Из таблицы 6 и рисунка 12 видно, что ухудшение качества полученного ОГА решения всего на 0,5 % от оптимального приводит к сокращению времени получения рационального решения в 3–6 раз по сравнению со временем получения оптимального решения. При этом с ростом размерности

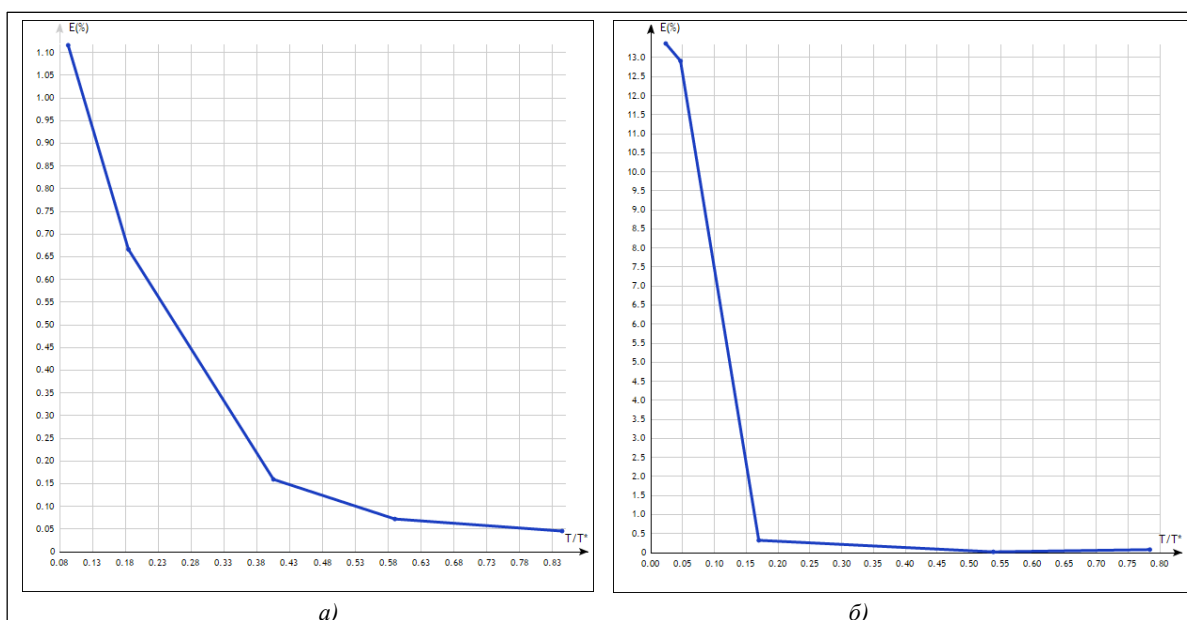


Рис. 12. Оценка влияния точности решения на время решения: а) задача 100×4; б) задача 150×4 (T* – время получения оптимального решения)

Fig. 12. Assessment of the impact on the accuracy of the solution and the solution time: а) the task 100×4; б) the task 150×4 (T* – time optimal solutions)

решаемой задачи данный эффект становится более явным. Это обстоятельство позволяет сделать вывод о перспективности и целесообразности применения ОГА для оперативного получения рационального (квазиоптимального) решения в условиях жестких временных ограничений.

Таким образом, результаты экспериментальной проверки показали, что применение ОГА для решения задач обеспечения устойчивости функционирования РИС эффективно для задач большой размерности (с числом переменных более 150), задач с жесткостью ограничений в интервале 40–50 %, а также в условиях жестких временных ограничений. В качестве схемы отбора особой целесообразно использовать случайный отбор или адаптивную схему. Использование простого ГА целесообразно для решения задач малой размерности (число переменных до 70). Использование ОГА для получения рационального решения поставленных задач позволит сократить время решения задачи размерности 100×4 по сравнению с применением точных методов в 5 раз с обеспечением размера относительной ошибки в 0,7 % и в 6–10 раз с обеспечением величины относительной ошибки в 1,5 %. С ростом размерности решаемых задач выигрыш по времени получения рационального решения становится еще более существенным.

Метод ветвей и границ целесообразно применять для решения задач обеспечения устойчивости функционирования РИС на этапах их проектирования, когда необходимо получить оптимальное решение и отсутствуют строгие ограничения на время его получения. На этапе эксплуатации, при реконфигурировании системы передачи и обработки данных, в условиях жесткого лимита времени на время реакции системы на отказы и сбои целесообразно применять метод получения рационального (квазиоптимального) решения, основанный на использовании ОГА, который в отличие от существующих приближенных методов обеспечивает за счет обоснованного выбора параметров алгоритма гарантированное получение рационального (квазиоптимального) решения (серии рациональных решений) за время, не превышающее заданное.

Литература

1. Воронин А.А., Морозов Б.И. Надежность информационных систем. СПб: Изд-во СПбГТУ, 2001. 89 с.
2. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. 2-е изд. СПб: Питер, 2004. 705 с.
3. Николаев В.И., Брук В.М. Системотехника: методы и приложения. Л.: Машиностроение, 1985. 199 с.
4. Есиков Д.О. Обеспечение устойчивого функционирования вычислительных систем за счет обеспечения высоконадежного хранения данных // Фундаментальные и прикладные исследования, разработка и применение высоких технологий в промышленности: сб. стат. XV Междунар. науч.-практич. конф. СПб: Изд-во Политехн. ун-та, 2013. С. 110–118.
5. Повзнер Л.Д. Теория систем управления. М.: Изд-во МГГУ, 2002. 472 с.
6. Данилкин Ф.А., Есиков Д.О. Оптимизация распределения информационных ресурсов в вычислительной сети // Инновации в технологиях и образовании: сб. стат. V Междунар. науч. конф. Белово: Изд-во КузГТУ, 2012. Ч. 2. 328 с.
7. Киселев В.Д., Есиков О.В., Кислицын А.С. Теоретические основы оптимизации информационно-вычислительного процесса и состава комплексов средств защиты информации в вычислительных сетях; [под ред. Е.М. Сухарева]. М.: Полиграфсервис XXI век, 2003. 198 с.
8. Есиков Д.О. Задачи обеспечения устойчивости функционирования распределенных информационных систем // Программные продукты и системы. 2015. № 4. С. 133–142.
9. Сергиенко И.В., Гуляницкий Л.Ф., Сиренко С.И. Классификация прикладных методов комбинаторной оптимизации // Кибернетика и системный анализ. 2009. № 5. С. 71–83.
10. Алексеев О.Г. Комплексное применение методов дискретной оптимизации. М.: Наука, 1987. 248 с.
11. Есиков Д.О. Оценка влияния предварительного определения порядка ветвления переменных на эффективность метода ветвей и границ // Междисциплинарные исследования в области математического моделирования и информатики: матер. науч.-практич. конф. Ульяновск: SIMJET, 2013. С. 17–21.
12. Бабаев А.А. Организация поиска решений на деревьях детерминированной структуры // Электронное моделирование. 1985. Т. 7. № 1. С. 19–25.
13. Есиков Д.О. Повышение эффективности метода ветвей и границ выбором способа организации компьютерных вычислений // Интеллектуальные и информационные системы (Интеллект 2013): сб. стат. Всерос. науч.-практич. конф. Тула: Изд-во ТулГУ, 2014. С. 143–147.
14. Yesikov D.O. Increase of level of stability of functioning of systems of storage and data processing at the expense of realization of actions for ensuring safety of information. Proc. 3rd Intern. Conf. High Performance Computing (HPC-UA 2013). Ukraine, Kiev, 2013, pp. 416–420.
15. Есиков О.В., Есиков Д.О. Математические модели и алгоритмы обеспечения сохранности информации // Изв. Тульского гос. ун-та: Технические науки. 2013. Вып. 9. Ч. II. С. 110–118.
16. Есиков Д.О., Ивутин А.Н. Программа решения задач целочисленного линейного программирования с булевыми переменными. Свид. о гос. регистр. прог. для ЭВМ № 2014619322. Госрегистр. в Реестре прог. для ЭВМ 15.09.2014.
17. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы; [пер. с пол. И.Д. Рудинского]. М.: Горячая линия–Телеком, 2004. 383 с.
18. Holland J.H. Adaptation in Natural and Artificial Systems: 2nd ed. Cambridge, MA: MIT Press, 1992.
19. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: Физматлит, 2006. 320 с.
20. Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Труды ИСА РАН. 2008. № 33. С. 206–219.
21. Есиков Д.О., Ивутин А.Н., Ларкин Е.В. Программа решения задач целочисленного программирования с булевыми переменными островным генетическим алгоритмом. Свид. о гос. регистр. прог. для ЭВМ № 2015663468. Госрегистр. в Реестре прог. для ЭВМ 18.12.2015.
22. Yesikov D.O., Ivutin A.N. Rational values of parameters of island genetic algorithms for the effective solution of problems of ensuring stability of functioning of the distributed information systems. Proc. 5th Mediterranean Conf. on Embedded Computing (MECO) 2016, pp. 309–312. DOI: 10.1109/MECO.2016.7525769.
23. Карпенко А.П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов // Информационные технологии. 2012. № 7. С. 13–15.

**EVALUATING THE EFFECTIVENESS OF SUSTAINABILITY PROBLEM SOLVING METHODS
OF DISTRIBUTED INFORMATION SYSTEM FUNCTIONING**

D.O. Esikov¹, *Postgraduate Student, mcgeen4@gmail.com*

¹ *Tula State University, Lenina Ave. 92, Tula, 300012, Russian Federation*

Abstract. To make informed decisions regarding the organization of data storage and processing to ensure the sustainability of distributed information systems the paper proposes to use a complex of mathematical models for optimizing the distribution of functional task software elements over network nodes; for optimizing the distribution of information resources by data storage and processing centers; for optimizing technical means of data storage and processing system; for optimizing the allocation of resources by information storage and processing centers. It is shown that these problems are related to the class of optimization problems with discrete Boolean variables.

The paper proposes and experimentally verifies the branch-and-bound method and the method of genetic algorithms to solve formalized problems. In order to improve the efficiency of the branch-and-bound method the paper proposes preliminary determination of the variable branching order using an approximation method for solving the dual problem once. The article shows an experimental verification of the effectiveness of the branch-and-bound method for solving problems of ensuring distributed information system sustainability including the use of the algorithm of predetermined order of branching variables. The author estimates the initial data influence on overall performance of the branch-and-bound method. The paper shows the most effective strategies of variables branching to solve the designed tasks. There are variations of major operators, as well as initialization circuits of a genetic algorithm initial population to solve the problems of ensuring distributed information system sustainability.

To improve the quality of the solution, which is obtained using a genetic algorithm, the author justifies the use of an adaptive scheme of individuals' reproduction and an computational island circuit. He also experimentally checks the effectiveness of the proposed genetic and island genetic algorithms. The paper determines the parameters of genetic algorithms for ensuring maximum quality of the solutions and proves ability to manage the accuracy of the obtained solution by changing the algorithm parameters when introducing time restrictions for solving. There is a comparative assessment of the branch-and-bound method and the island genetic algorithm for solving formal problems. The paper also shows areas of their effective application.

Keywords: distributed information system, operation stability, mathematical models, branch-and-bound method, dual problem, genetic algorithm, island genetic algorithm, adaptive reproduction scheme.

References

1. Voronin A.A., Morozov B.I. *Nadezhnost informatsionnykh sistem* [Information System Reliability]. St. Petersburg, SPbPU Publ., 2001, 89 p.
2. Broido V.L. *Vychislitelnye sistemy, seti i telekommunikatsii* [Computer Systems, Networks and Telecommunications]. Piter Publ., 2nd ed., 2004, 705 p.
3. Nikolaev V.I., Brooke V.M. *Sistemotekhnika: metody i prilozheniya* [System engineering: methods and applications]. L., Mashinostroenie Publ., 1985, 199 p.
4. Yesikov D.O. Sustainable operation of computer systems by providing highly reliable data storage. *Fundamentalnye i prikladnye issledovaniya, razrabotka i primeneniye vysokikh tekhnologiy v promyshlennosti: sb. stat. XV Mezhdunar. nauch.-praktich. konf.* [Proc. 15 Int. Science and Practice Conf. "Fundamental and Applied Research, Development and Application of High Technologies in the Industry"]. St. Petersburg, Polytechnic Univ. Publ., 2013, pp. 110–118 (in Russ.).
5. Povzner L.D. *Teoriya sistem upravleniya* [Theory of Control Systems]. Moscow, MSMU Publ., 2002, 472 p.
6. Danilkin F.A., Yesikov D.O. Optimization of informational resources distribution in a computer network. *Innovatsii v tekhnologiyakh i obrazovanii: sb. stat. V Mezhdunar. nauch. konf.* [Proc. 5th Int. Scientific Conf. "Innovations in Technologies and Education"]. KuzGTU branch in Belovo Publ., 2012, part 2, 328 p.
7. Kiselev V.D., Esikov O.V., Kislitsyn A.S. *Teoreticheskie osnovy optimizatsii informatsionno-vychislitelnogo protsessa i sostava kompleksov sredstv zashchity informatsii v vychislitelnykh setyakh* [Theoretical bases of information and computing process optimization and structure of information security systems in computer networks]. E.M. Sukharev (Ed.). Moscow, Poligrafservis XXI vek Publ., 2003, 198 p.
8. Yesikov D.O. The problems of ensuring distributed information systems sustainability. *Programmnye produkty i sistemy* [Software and Systems]. 2015, no. 4 (112), pp. 133–142 (in Russ.).
9. Sergienko I.V., Gulyanitsky L.F., Sirenko S.I. Classification of combinatorial optimization applied methods. *Kibernetika i sistemnyy analiz* [Cybernetics and Systems Analysis]. 2009, no. 5, pp. 71–83.
10. Alekseev O.G. *Kompleksnoe primeneniye metodov diskretnoy optimizatsii* [Complex Application of Discrete Optimization Methods]. Moscow, Nauka Publ., 1987, 248 p.
11. Yesikov D.O. Assessing the impact of pre-determining the order of the branching variables on the efficiency of branch and bound method. *Mezhdistsiplinarnye issledovaniya v oblasti matematicheskogo modelirovaniya i informatiki: mater. nauch.-praktich. konf.* [Proc. Science and Practice Conf. "Interdisciplinary research in the field of mathematical modeling and computer science"]. Ulyanovsk, SIMJET Publ., 2013, pp. 17–21 (in Russ.).
12. Babaev A.A. Organization of Decision Search on Deterministic Structure Trees. *Elektronnoye modelirovaniye* [Electronic modeling]. 1985, vol. 7, no. 1, pp. 19–25 (in Russ.).

13. Yesikov D.O. Improving the efficiency of branch-and-bound method selection mode of organization of computing. *Intellectualnye i informatsionnye sistemy (Intellect 2013): sb. stat. Vseros. nauch.-praktich. konf.* [Proc. All-Russian Science and Practice Conf. "Intellectual and information systems. Intellect 2013"]. Tula, Tula State Univ. Publ., 2014, pp. 143–147 (in Russ.).
14. Yesikov D.O. Increase of level of stability of functioning of systems of storage and data processing at the expense of realization of actions for ensuring safety of information. *Proc. 3rd Int. Conf. "High Performance Computing" (HPC-UA 2013)*. Ukraine, Kyiv, 2013, pp. 416–420.
15. Yesikov O.V., Yesikov D.O. Mathematical models and algorithms to ensure data safety. *Izv. Tul'skogo gos. un-ta: Tekhnicheskie nauki* [News of the Tula State Univ. Engineering Sciences]. 2013, iss. 9, part 2, Tula, pp. 109–118 (in Russ.).
16. Esikov D.O., Ivutin A.N. *Programma resheniya zadach tselochislennogo lineynogo programmirovaniya s bulevymi peremennymi* [A Problem-solver for integer linear programming problems with Boolean variables]. State registration Certificate of the computer program № 2014619322. 2014 (in Russ.).
17. Rutkowski J., Pilinsky M., Rutkowski L. *Neural networks, genetic algorithms and fuzzy systems*. (Russ.ed.: I.D. Rudinsky. Moscow, Goryachaya Liniya–Telecom Publ., 2004, 383 p.).
18. Holland J.H. *Adaptation in Natural and Artificial Systems*. 2nd ed. Cambridge, MA, MIT Press, 1992.
19. Gladkov L.A., Kureychik V.V., Kureychik V.M. *Geneticheskie algoritmy* [Genetic algorithms]. Moscow, Fizmatlit Publ., 2006, 320 p.
20. Nguyen M.H. Application of a genetic algorithm to the set coating finding problem. *Trudy ISA RAN* [Proc. ISA RAS]. 2008, no. 33, pp. 206–219 (in Russ.).
21. Esikov D.O., Ivutin A.N., Larkin E.V. *Programma resheniya zadach tselochislennogo programmirovaniya s bulevymi peremennymi ostrovnym geneticheskim algoritmom* [A Program for Solving Integer Programming Problems with Boolean Variables Using an Island Genetic Algorithm]. State registration Certificate of the computer program № 2015663468. 2015 (in Russ.).
22. Yesikov D.O., Ivutin A.N. Rational values of parameters of island genetic algorithms for the effective solution of problems of ensuring stability of functioning of the distributed information systems. *Proc. 5th Mediterranean Conf. on Embedded Computing (MECO)*. 2016, pp. 309–312.
23. Karpenko A.P. Population algorithms of global search optimization. Overview of new and little-known algorithms. *Informatsionnye tekhnologii* [Information Technologies]. 2012, no. 7, pp. 13–15 (in Russ.).

Примеры библиографического описания статьи

1. Есиков Д.О. Оценка эффективности методов решения задач обеспечения устойчивости функционирования распределенных информационных систем // Программные продукты и системы. 2017. Т. 30. № 2. С. 241–256; DOI: 10.15827/0236-235X.118.241-256.
2. Esikov D.O. Evaluating the effectiveness of sustainability problem solving methods of distributed information system functioning. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 241–256 (in Russ.); DOI: 10.15827/0236-235X.118.241-256.

УДК 620.1

DOI: 10.15827/0236-235X.118.257-260

Дата подачи статьи: 09.09.16

2017. Т. 30. № 2. С. 257–260

РЕКУРСИВНЫЙ АЛГОРИТМ ТОЧНОГО РАСЧЕТА РАНГОВЫХ КРИТЕРИЕВ ПРОВЕРКИ СТАТИСТИЧЕСКИХ ГИПОТЕЗ

*А.В. Агамиров, д.т.н., профессор, mmk@mati.ru
(Национальный исследовательский университет «МЭИ»,
ул. Красноказарменная, 14, г. Москва, 111250, Россия);*

*В.А. Вестяк, к.ф.-м.н., доцент, v.a.vestyak@mail.ru
(Московский авиационный институт (национальный исследовательский университет),
Волоколамское шоссе, 4, г. Москва, 125993, Россия);*

В.А. Агамиров, к.т.н., аналитик, avl095@mail.ru

В статье рассматривается методика генерации точных распределений ранговых непараметрических критериев средствами компьютерной комбинаторики.

Актуальность работы обусловлена затруднениями в определении точных распределений критических значений ранговых критериев проверки статистических гипотез из-за того, что точные таблицы, рекуррентные формулы для многих критериев не существуют, а аппроксимации часто дают неудовлетворительный результат при ограниченных объемах наблюдений.

Задача расчета распределения ранговых критериев заключается в переборе всех возможных вариантов перестановок выборок и в расчете ранговых статистик, а также накопленных частот их появления.

Для ее решения разработана программа генерации перестановок элементов выборок ранговых непараметрических критериев, основанная на рекурсивном алгоритме прямого перебора вариантов перестановок вектора порядковых статистик со следующим ограничением числа вариантов: во всех вариантах перестановок элементы одной и той же выборки не могут меняться местами, что является универсальным условием для всех точных распределений ранговых критериев.

В работе приводится ссылка на интернет-ресурс, содержащий программный комплекс реализации алгоритма расчета ранговых критериев. В данном комплексе рассмотрены четыре непараметрических критерия: двухвыборочный критерий Уилкоксона, критерий Лемана–Розенблатта, критерий серий и критерий Краскала–Уоллиса, точные распределения статистик которых представляют наибольший интерес для технических задач. Рассматриваемый алгоритм может быть использован и для других ранговых критериев проверки статистических гипотез.

В работе представлена разработанная авторами реализация метода генерации точных распределений ранговых непараметрических критериев средствами компьютерной комбинаторики, основанная на рекурсивном прямом переборе вариантов перестановок вектора порядковых статистик с последующей фильтрацией результатов. Таким образом, решена задача определения критических значений ранговых непараметрических критериев для проверки статистических гипотез.

Ключевые слова: непараметрические ранговые критерии, проверка гипотез, точное распределение, комбинаторика, алгоритм, программа, JavaScript.

Задача проверки статистических гипотез во всех случаях сопряжена с необходимостью определения критических значений критериев. В то же время для большинства ранговых критериев определение точных распределений является весьма непростой задачей как с математической, так и с вычислительной точки зрения. Различного рода аппроксимации зачастую дают неудовлетворительный результат при ограниченных объемах наблюдений, свойственных анализу данных в технических задачах, связанных со значительным рассеянием свойств, вследствие структурной неоднородности конструкционных материалов и большой вариативности внешних факторов при проведении испытаний. Точные таблицы, рекуррентные формулы, производящие функции частот и моментов для многих критериев не существуют [1–4]. Кроме того, при современном развитии вычислительной техники более предпочтительным является точный компьютерный расчет. Подробнее эти вопросы обсуждаются в [5], в данной работе предлагается методика генерации точных распределений ранговых

непараметрических критериев средствами компьютерной комбинаторики.

С вычислительной точки зрения распределение ранговых критериев представляет собой перебор всех возможных вариантов перестановок элементов выборочных совокупностей при некоторых граничных условиях с последующим расчетом ранговых статистик и накопленных частот их появления. Предлагаемый далее алгоритм применим для большинства критериев, для которых вычисляются выборочные ранговые статистики. Для некоторых критериев существуют более эффективные методы [5, 6], однако с целью обобщения здесь рассматриваются различные критерии независимо от наличия иных методов расчета точных распределений. Авторы сознают, что предлагаемый алгоритм имеет недостаток – перебор большого числа лишних вариантов, что зачастую, ведет к существенному увеличению машинного времени. Так что, данный алгоритм может рассматриваться как некоторый шаг в направлении применения методов рекурсивной

компьютерной комбинаторики в задачах статистического анализа. К тому же при постоянном увеличении производительности процессоров современных компьютеров проблема машинного времени постепенно отступает на второй план.

В рассматриваемой ниже программе, написанной на языке Javascript, используется метод генерации перестановок элементов выборок ранговых непараметрических критериев. Метод основан на рекурсивном прямом переборе вариантов перестановок вектора порядковых статистик объема n с последующей фильтрацией результатов с учетом универсального для всех точных распределений ранговых критериев условия: внутри данной выборки порядковые номера элементов должны сохраняться независимо от места их расположения. Естественно, что за пределами выборки порядковые номера элементов должны принимать все возможные варианты перестановок.

Для примера рассмотрим возможные варианты таких перестановок для трех выборок ($k=3$) объемами $n_1 = 1, n_2 = 1, n_3 = 2$.

При суммарном объеме выборок $N = \sum_{i=1}^k n_i = 4$

безусловное число вариантов перестановок элементов равно $N!$, то есть 24. С учетом указанного выше условия число таких вариантов сокращается

$$\text{до } 12: kk = \frac{N!}{\prod_{i=1}^k n_i!} = \frac{4!}{1!1!2!} = 12.$$

В таблице перечислены эти варианты, выборки обозначены символами X, Y, Z.

Варианты перестановок рангов трех выборок объемами $n_1=1, n_2=1, n_3=2$

Permutations of ranks of 3 samples with a volume of $n_1=1, n_2=1, n_3=2$

Номер варианта	Перестановка	Вектор a	Вектор $aindex$
1	X ₁ Y ₁ Z ₁ Z ₂	1 2 3 4	1 2 3 3
2	X ₁ Z ₁ Y ₁ Z ₂	1 3 2 4	1 3 2 3
3	X ₁ Z ₁ Z ₂ Y ₁	1 3 4 2	1 3 3 2
4	Y ₁ X ₁ Z ₁ Z ₂	2 1 3 4	2 1 3 3
5	Y ₁ Z ₁ X ₁ Z ₂	2 3 1 4	2 3 1 3
6	Y ₁ Z ₁ Z ₂ X ₁	2 3 4 1	2 3 3 1
7	Z ₁ X ₁ Y ₂ Z ₂	3 1 2 4	3 1 2 3
8	Z ₁ X ₁ Z ₂ Y ₁	3 1 4 2	3 1 3 2
9	Z ₁ Y ₁ X ₁ Z ₂	3 2 1 4	3 2 1 3
10	Z ₁ Y ₁ Z ₂ X ₁	3 2 4 1	3 2 3 1
11	Z ₁ Z ₂ X ₁ Y ₁	3 4 1 2	3 3 1 2
12	Z ₁ Z ₂ Y ₁ X ₁	3 4 2 1	3 3 2 1

Отметим еще раз, что номера 3 и 4, обозначающие элементы третьей выборки объемом 2, во всех вариантах сохраняют порядок, что и является условием генерации точного распределения для любого рангового критерия, в то время как порядковые номера первой и второй выборок, имеющих единичные объемы, могут меняться местами.

Далее приведена программа генерации перестановок при наличии граничных условий.

```

1. <html>
2. <script>
3. var kk;
4. kk=0;
5. stat_crit();
//*****
6. function stat_crit() {
7. var kx; // количество выборок
8. var a=[]; // вектор порядковых номеров элементов выборок
9. var aindex=[]; //вспомогательный вектор индексов выборок
10. var m=[]; //вектор объемов выборок
11. var km;
12. var n; //суммарный объем выборок
13. kx=3;n=0;m[1]=1;m[2]=2;m[3]=2;
14. for (i=1;i<=kx;i++) n=n+m[i];
15. for (i=1;i<=n;i++) a[i]=i;
16. km=0;
17. for (i=1;i<=kx;i++) {
18. for (j=1;j<=m[i];j++) aindex[j+km]=i;
19. km=km+m[i];
20. }
21. omega(a,aindex,n,n);
22. }
//*****
23. function omega(a,aindex,n,mx) {
//рекурсивная функция генерации перестановок от 1 до n
24. var i,j,ss;
25. if (mx==1) {
// функция фильтрации и вывода на печать текущей перестановки
// элементов вектора a[1..n]
26. filter(true,n,a,aindex); }
27. else {
28. for (i=1;i<=mx;i++) {
29. ss=a[1];
30. for (j=1;j<=mx-1;j++) a[j]=a[j+1];
31. a[mx]=ss;
32. omega(a,aindex,n,mx-1);
33. }
34. }
35. }
//*****
// функция фильтрации и печати текущей перестановки из
// kk=n!/((m[1]!*m[2]!*...*m[kx]!)) вариантов
36. function filter(fltr,n,a,aindex) {
37. test=0;
38. if(fltr) {
39. if (test==0) {
40. for (i=1;i<=n;i++) {
41. for (j=i;j<=n;j++) {
//отмена вывода перестановки элементов при нарушении их
// порядковых номеров внутри выборки
42. if (aindex[a[i]]==aindex[a[j]] && a[i]>a[j]) {test=1;break;}
43. }
44. if (test==1) break;
45. }
46. }
47. }
48. if (test==0) {
49. kk++;
50. document.write(kk+" ");
// вывод на печать вектора a[1..n]
51. for (i=1;i<=n;i++) document.write(a[i]+" ");
52. document.write("<br>");
53. }
54. }
//*****
55. </script>
56. </html>

```

Программа содержит три основные функции. Первая функция – `stat_crit` (строки 6–20) формирует исходные данные рассмотренного выше примера и два вспомогательных вектора a и $aindex$ размерности N , первый из которых представляет собой вектор порядковых номеров объединенной выборки, второй – вектор, содержащий номера выборок, как это показано в таблице.

Вторая функция – `omega(a, aindex, n, n)` (строки 23–35) является рекурсивной функцией генерации перестановок элементов от 1 до n , в строке 32 которой представлена процедура рекурсии с уменьшением на единицу объема объединенной выборки от n до 1. В этом случае (строка 26) реализуется фильтрация перестановок в соответствии с принятыми граничными условиями, для чего введена третья функция – `filter(fltr, n, a, aindex)` (строки 36–54). Отметим, что фильтрация производится при значении параметра `fltr=true` (строка 38), в противном случае выводятся все варианты перестановок. На выходе этой функции выводятся элементы вектора a с учетом граничных условий, как показано в таблице.

Все остальные действия, необходимые для расчета статистик ранговых критериев, являются достаточно тривиальной вычислительной задачей и в данной работе не обсуждаются.

Полная версия программы с открытым кодом размещена на Javascript по ссылке http://inteh.mpei.ru/programs/stat/Menu/stat_tree.html. В данном комплексе рассмотрены четыре непараметрических критерия: двухвыборочный критерий Уилкоксона [1–4], критерий серий [7], критерий Лемана–Розенблатта [8] и критерий Краскела–Уоллиса [3, 4], точные распределения статистик которых, по мнению авторов, представляют наибольший интерес, по крайней мере, для технических задач. Очевидно, что представленный алгоритм может быть использован и для других ранговых критериев проверки статистических гипотез.

Для полноты картины рассмотрим указанные критерии, за исключением критерия серий, распределение статистики которого не требует вычисления рангов, а может быть подсчитано [7, 9] более простым способом, что и реализовано в полной версии программного комплекса.

Двухвыборочный критерий Уилкоксона предназначен для проверки гипотезы об отсутствии сдвига двух независимых выборок, то есть об отсутствии различия между медианами двух совокупностей при одинаковом, но произвольном распределении [1, 2]. Пусть x_1, x_2, \dots, x_m – случайная выборка из $F(x-\theta_x)$, y_1, y_2, \dots, y_n – случайная выборка из $F(y-\theta_y)$ ($m \leq n$). Функцию распределения F не предполагают симметричной, но форма распределения должна быть одинаковой для двух совокупностей. Для проверки нулевой гипотезы о том, что обе выборки извлечены из одной и той же совокупности $H_0: \Delta = \theta_y - \theta_x = 0$ против альтернативы $H_A: \Delta \neq 0$ строят вариационный ряд из $k = m + n$

наблюдений и присваивают им ранги, равные порядковому номеру наблюдения в общем вариационном ряду. Далее рассчитывают сумму рангов меньшей выборки в общем вариационном ряду:

$$W = \sum_{i=1}^m R_i.$$

Для проверки нулевой гипотезы $H_0: \Delta = 0$ при альтернативной гипотезе $H_A: \Delta < 0$ должно выполняться неравенство $W > W_{\alpha l}$. При альтернативной гипотезе $H_A: \Delta > 0$ должно выполняться неравенство $W \leq W_{\alpha u}$. При двусторонней альтернативной гипотезе $H_A: \Delta \neq 0$ должно выполняться неравенство $W_{\alpha l} \leq W \leq W_{\alpha u}$ с уровнем значимости 2α .

Критерий Лемана–Розенблатта проверяет гипотезу об однородности двух выборок. Проверяется нулевая гипотеза о том, что две выборки извлечены из одной и той же генеральной совокупности, то есть $H_0: F(x) = G(x)$ при любом x . Статистика критерия рассчитывается по формуле

$$\omega^2 = \frac{1}{n \cdot m} \left[\frac{1}{6} + \frac{\sum_{i=1}^n (R_i - i)^2}{m} + \frac{\sum_{j=1}^m (S_j - j)^2}{n} \right] + \frac{2}{3},$$

где R_i, S_j – ранги первой выборки объемом n и второй объемом m в общем вариационном ряду. Нулевую гипотезу принимают, если $\omega^2 \leq \omega_{\alpha}^2$ с уровнем значимости α . В противном случае принимают альтернативную гипотезу.

Критерий Краскела–Уоллиса обобщает задачу о двух выборках на случай k выборок: x_{ij} , $i = 1, \dots, k, j = 1, \dots, n_j$, с функциями распределения $F(x - \theta_j)$, где n_j – число наблюдений в j -й выборке. Нулевая гипотеза утверждает, что k выборок из произвольных совокупностей можно рассматривать как одну (объединенную) выборку из общей совокупности, то есть подтверждается равенство параметров сдвига θ_j , когда не задано значение общего параметра масштаба $H_0: \theta_1 = \theta_2 = \dots = \theta_k$ против альтернативы $H_A: \theta_1, \dots, \theta_k$ не все равны. Для проверки нулевой гипотезы строят общий вариационный ряд из $n = \sum_{i=1}^k n_i$ наблюдений и рассчитывают статистику:

$$H = \frac{12}{(n+1)} \cdot \sum_{i=1}^k \frac{R_i^2}{n_i} - 3 \cdot (n+1),$$

где R_i – сумма рангов i -й выборки в общем вариационном ряду.

Нулевую гипотезу принимают, если $H \leq H_{\alpha}$ с уровнем значимости α . В противном случае принимают альтернативную гипотезу.

Другим способом проверки выборочной гипотезы k является попарное сравнение выборок по критерию Уилкоксона с вычислением точных критических значений.

Таким образом, разработанный алгоритм, являющийся реализацией метода рекурсивной компью-

терной комбинаторики, может быть использован для решения задачи проверки статистических гипотез, связанной с определением критических значений критериев. Учитывая, что для большинства ранговых критериев определение точных распределений является сложной задачей с математической и вычислительной точек зрения, представленная в работе методика генерации точных распределений критериев средствами компьютерной комбинаторики является эффективным решением.

Литература

1. Кендалл М.Дж., Стьюарт А. Теория распределений. М.: Наука, 1966.

2. Кендалл М.Дж., Стьюарт А. Статистические выводы и связи. М.: Наука, 1973. 899 с.

3. Хеттманспергер Т. Статистические выводы, основанные на рангах. М.: Финансы и статистика, 1987. 334 с.

4. Холлендер М., Вулф Д. Непараметрические методы статистики. М.: Финансы и статистика, 1983. 518 с.

5. Агамиров Л.В., Агамиров В.Л., Вестяк В.А. Численные методы и алгоритмы расчета точных распределений непараметрических критериев проверки статистических гипотез // Вестн. МАИ. 2013. Т. 20. № 4. С. 212–218.

6. Агамиров Л.В. Методы статистического анализа механических испытаний. М.: Интернет Инжиниринг, 2004. 127 с.

7. Шуленин В.П. Математическая статистика. Ч. 2: Непараметрическая статистика. Томск: Изд-во НТЛ, 2012. 388 с.

8. Леман Э. Проверка статистических гипотез. М.: Наука, 1964. 498 с.

9. Липский В. Комбинаторика для программистов. М.: Мир, 1988. 200 с.

Software & Systems

DOI: 10.15827/0236-235X.118.257-260

Received 09.09.16

2017, vol. 30, no. 2, pp. 257–260

RECURSIVE ALGORITHM FOR EXACT CALCULATION OF RANK TESTS FOR TESTING STATISTICAL HYPOTHESES

L.V. Agamirov¹, Dr.Sc. (Engineering), Professor, mmk@mati.ru

V.A. Vestyak², Ph.D. (Physics and Mathematics), Associate Professor, v.a.vestyak@mail.ru

V.L. Agamirov, Ph.D. (Engineering), analyst, avl095@mail.ru

¹ National Research University "MPEI", Krasnokazarmennaya St. 14, Moscow, 111250, Russian Federation

² Moscow Aviation Institute (National Research University), Volokolamskoe Highway 4, Moscow, 125993, Russian Federation

Abstract. The paper considers the method of generating exact distributions of nonparametric rank tests by means of the computer combinatorial theory.

Relevance of the work consists in the fact that determination of exact distribution of critical values of rank tests for statistical hypotheses testing is complicated by the fact that the exact tables and recurrence formulas for many of the tests do not exist. In addition, approximations often give unsatisfactory results at limited volumes of observations.

The task of calculating the distribution for rank tests is a search of all possible sample permutations and calculations of rank statistics, as well as cumulative frequency of their occurrence.

The program of generating permutations of elements of samples for nonparametric rank criteria based on the recursive brute-force algorithm of direct enumeration of order statistics vector permutation is developed with the following limited number of options: in all permutation options the elements from the same sample cannot be swapped. It is a universal condition for all the rank criteria exact distributions.

The paper refers to the Internet resource that contains the software package implementation of the considered calculation algorithm for a rank test. This complex contains four nonparametric criteria: two-sample Wilcoxon test, Lehmann-Rosenblatt test, series test and Kruskal-Wallis test, whose accurate distribution statistics are of greatest interest for technical problems. The algorithm can be used for other rank tests of statistical hypotheses testing.

The paper presents an implementation of the generation method of nonparametric rank test exact distributions by computer combinatorial means. It is based on the developed by the authors recursive direct enumeration of options of order statistics vector permutation with following filtration of the results. Thus, the authors solve the problem of determining the critical values of nonparametric rank tests for testing statistical hypotheses.

Keywords: nonparametric rank tests, hypothesis testing, exact distribution, combinatorics, algorithm, program, JavaScript.

References

1. Kendall M.G., Stuart A. *The Advanced Theory of Statistics. Vol. 1. Distribution Theory*. Charles Griffin Publ., 4th ed., 1952 (Russ. ed.: Moscow, Nauka Publ., 1966).

2. Kendall M.G., Stuart A. *The Advanced Theory of Statistics. Vol. 1. Inference and Relationship*. Hafner Publ. Company, 3rd ed., 1961, 676 p. (Russ. ed.: Moscow, Nauka Publ., 1973, 899 p.).

3. Hettmansperger T.P. *Statistical Inference Based on Ranks*. NY, J. Wiley and Sons Publ., 1984 (Russ. ed.: Moscow, Finansy i statistika Publ., 1987, 334 p.).

4. Hollander M., Wolfe D. *Nonparametric Statistical Methods*. Wiley Publ., 1999 (Russ. ed.: Moscow, Finansy i statistika Publ., 1983, 518 p.).

5. Agamirov L.V., Agamirov V.L., Vestyak V.A. Numerical methods and algorithms of calculation of exact distributions of nonparametrical criteria statistical hypotheses. *Vestnik MAI* [Vestnik Moskovskogo Aviatcionnogo Instituta]. 2013, vol. 20, no. 4, pp. 212–218 (in Russ.).

6. Agamirov L.V. *Metody statisticheskogo analiza mekhanicheskikh ispytany* [The Methods of Mechanical Test Statistical Analysis]. Moscow, Internet Inzhiniring Publ., 2004, 127 p.

7. Shulenin V.P. *Matematicheskaya statistika* [Mathematical Statistics]. Tomsk, NTL Publ., 2012, 388 p.

8. Lehman E.L. *Testing Statistical Hypotheses*. J. Wiley & Sons Publ., NY, 1959 (Russ. ed.: Moscow, Nauka Publ., 1964, 498 p.).

9. Lipsky B. *Kombinatorika dlya programmistov* [Combinatorics for Programmers]. Moscow, Mir Publ., 1988, 200 p.

УДК 004.89 + 004.91

DOI: 10.15827/0236-235X.118.261-269

Дата подачи статьи: 13.03.17

2017. Т. 30. № 2. С. 261–269

ПРОГРАММНЫЙ КОМПЛЕКС РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ*А.С. Григораш, аспирант, grigoraschandreya@gmail.com;**В.М. Курейчик, д.т.н., зав. кафедрой, kur@tgn.sfedu.ru;**В.В. Курейчик, д.т.н., зав. кафедрой, vkur@sfedu.ru**(Южный федеральный университет, пер. Некрасовский, 44, г. Таганрог, 347922, Россия)*

В последнее время информационные технологии неуклонно развиваются. В связи с этим стремительно увеличиваются объемы банков данных, обрабатывать которые с каждым днем становится все сложнее. Актуальным направлением обработки больших массивов данных является их интеллектуальный анализ, методы которого позволяют обнаруживать в наборах данных неявные закономерности. Одна из основных задач интеллектуального анализа данных – задача кластеризации. Авторами статьи сформулирована ее постановка. Кластеризация является NP-полной, трудной задачей, поэтому необходимо разрабатывать нестандартные алгоритмы и методы для нахождения эффективного решения за полиномиальное время.

Цель данной работы – построение программного комплекса для решения задачи кластеризации. Особенностью предлагаемого решения является использование модифицированных методов эволюционного моделирования и роевого интеллекта, которые адаптируются к изменениям внешней среды.

Была выделена модификация для нахождения квазиоптимальных решений, позволившая значительно сократить время формирования кластеров. Разработаны модифицированные генетический, муравьиный и пчелиный алгоритмы, а также модель функционирования алгоритмов как единая система. Рассмотрена структура программного комплекса. Разработан интуитивный пользовательский интерфейс.

Был проведен вычислительный эксперимент на банках данных различного объема. В ходе серии вычислительных экспериментов время выполнения кластеризации элементов приблизительно равно 9.4 сек. В среднем полученное значение отличается от оптимального на 3–5 %. В рамках данной работы оптимальное значение времени равно 8–9 сек. Вычислительный эксперимент выявил временную сложность разработанного комплекса, которая приблизительно равна $O(n^2)$.

Ключевые слова: интеллектуальный анализ данных, эволюционное моделирование, генетический алгоритм, роевой интеллект, пчелиный алгоритм, муравьиный алгоритм.

Задача кластеризации – одна из главных задач интеллектуального анализа данных [1–4]. Она особенно значима, когда выступает одним из этапов анализа данных, построения законченного аналитического решения. Аналитику обычно легче выделить в группу схожие объекты, изучить их особенности и построить для каждой группы отдельную модель данных. Таким приемом постоянно пользуются для обработки банков данных [5–8].

Новизна разработанных методов и подходов к решению задачи кластеризации заключается в использовании методов, инспирированных природными системами: генетическим, муравьиным и пчелиным алгоритмами. Они позволяют ускорить процесс обработки банков данных. Применение таких подходов дает преимущество в решении NP-полных, трудных задач [9–12]. Рассмотрим постановку задачи кластеризации.

Постановка задачи кластеризации

Пусть X – множество объектов, Y – множество меток кластеров и задана функция расстояния между объектами:

$$\rho(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2}, \quad (1)$$

где x_i, x'_i – характеристики объектов. Тогда существует конечная обучающая выборка объектов:

$$X^m = \{x_1, \dots, x_m\} \subset X. \quad (2)$$

Эту выборку необходимо разбить на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый из них состоял из объектов, близких по метрике ρ , а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается метка кластера y_i .

Алгоритм кластеризации – это функция $a: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$.

Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определения оптимального числа кластеров с точки зрения того или иного критерия качества кластеризации. При решении задачи кластеризации метки исходных объектов y_i изначально не заданы, и даже само множество Y может быть неизвестно [1–4].

В рамках данной работы банк данных имеет величину в 500 элементов, полученных случайным образом (см. таблицу).

Банк элементов**A bank of elements**

Элемент	Характеристика			
	X	Y	Z	V
1	36	15	4	61
2	69	23	11	32
3	93	69	43	33
4	23	33	28	80
5	70	24	33	47
6	79	98	87	80
-	-	-	-	-
500	18	5	62	10

Требуется разбить его на N кластеров: G_1, G_2, G_3, \dots , но при этом уже существует некоторое разбиение, которое необходимо оптимизировать, минимизировав среднее внутреннее кластерное расстояние или целевую функцию (ЦФ):

$$F = \frac{\sum_{y < j} [y_i = y_j] p(x_i, x_j)}{\sum_{i < j} [y_i = y_j]} \rightarrow \min, \quad (3)$$

где x_i, y_i – характеристики центра кластера; x_j, y_j – усредненные характеристики объектов.

В данной работе для оптимального решения задачи кластеризации производится случайное разбиение на кластеры, которое представляет собой вектор (рис. 1). Длина вектора равна количеству элементов банка данных, а также n элементов со значением 0, означающих конец одного кластера или начало другого [9–12].

1	5	4	10	6	2	3	0	8	9	7	0	..	351	320	500
---	---	---	----	---	---	---	---	---	---	---	---	----	-----	-----	-----

Рис. 1. Случайное разбиение элементов на кластеры
Fig. 1. Random splitting of elements on clusters

Для подсчета ЦФ необходимо найти центр каждого кластера. Для этого подсчитываются усредненные значения каждой характеристики для кластера в целом:

$$A_{\text{ср}} = \frac{\sum A}{n}, \quad (4)$$

где A – значение характеристики; n – количество всех значений характеристики A .

На рисунке 2 приведены усредненные значения характеристик для каждого кластера в целом. После нахождения центров кластеров подсчитываем ЦФ по формуле (3) и получаем $\text{ЦФ}_1 = 50$, $\text{ЦФ}_2 = 43$, $\text{ЦФ}_n = 63$.

G_1	53	39	33	53
G_2	53	11	69	50
\dots	-	-	-	-
G_n	16	27	13	45

Рис. 2. Центры кластеров
Fig. 2. Cluster centers

Классические последовательные алгоритмы представляют один случайный элемент из одного кластера в другой и снова производят подсчет, что является трудоемким процессом. Действия повторяются до тех пор, пока кластеры не будут полностью заполнены, а ЦФ не станет квазимиимальной [13–15]. Рассмотрим алгоритмы, разработанные для оптимизации процесса разбиения на кластеры.

Описание алгоритмов и программного комплекса

Особенностью предложенного подхода является использование алгоритмов, моделирующих процесс живой природы. Разработанные алго-

ритмы моделируют различные аспекты эволюции [9–13, 16–20].

Генетический алгоритм (ГА) моделирует процесс образования хромосом путем скрещивания, мутации и отбора [10, 12, 13, 16]. Это основная составляющая в природе. Муравьиный и пчелиный алгоритмы являются методами роевого интеллекта, главная составляющая которого – моделирование поведения живых организмов при коллективном взаимодействии [10, 17–20]. Случайное разбиение на кластеры в алгоритмах, инспирированных природными системами, формирует хромосому (набор альтернативных решений).

Основная цель ГА – минимизировать число внутрикластерных связей между элементами или максимизировать число внешних связей кластеров. При решении оптимизационных задач ГА имеет ряд преимуществ. Одно из них – адаптация к изменяющейся внешней среде. В процессе решения поставленная задача может подвергаться изменениям. Использование традиционных методов приводит к большим затратам машинного времени.

Алгоритмы роевого интеллекта (муравьиный и пчелиный) описывают поведение в многоагентных системах [10, 17–20]. Цель пчелиного алгоритма – разбить предоставленную информацию на участки. Путем исследований с помощью агентов производится поиск эффективного решения в центрах полученных участков. Если решение было найдено в окрестностях данной точки, агенты отправляются для полного и точного поиска эффективных решений.

Муравьиный алгоритм моделирует поведение муравьев в колонии. Поиск участков пищи производится сообща группой агентов. Проходя некоторый путь, агент помечает его феромоном – отличительное свойство агентов не возвращаться в точку, в которой уже были. Путем накопления феромона после прохождения всей группы происходит формирование кластера, и решения, попавшие в замкнутый участок, являются эффективными.

Для реализации задачи кластеризации был создан программный комплекс, к основным параметрам которого относятся количество элементов и настройки разработанных алгоритмов. Размеры кластеров генерируются случайно. После того, как кластеры созданы (или загружены из файла), необходимо произвести настройку разработанных алгоритмов.

Генетический алгоритм

Для эффективного использования ГА необходимо определить метод генерации начальной популяции. Существует множество методов построения начальной популяции [10, 12, 13]. В данной работе используется решето Эратосфена. Это алгоритм нахождения всех простых чисел до некоторого целого числа n . Решето подразумевает фильтрацию всех чисел, за исключением простых.

По мере прохождения списка нужные числа остаются, а ненужные (они называются составными) исключаются. Рассмотрим модифицированные генетические операторы, используемые для решения задачи кластеризации.

Оператор кроссинговера. Процесс, при котором происходят сближение двух гомологичных хромосом при созревании половых клеток и обмен частями. Существуют различные модификации оператора кроссинговера. Из-за большой информационной нагрузки в данной работе используется упорядоченный оператор кроссинговера, при котором у родителей выбирается точка разрезания, относительно которой будет происходить копирование элементов. Выбираются элементы из левого сегмента первого родителя и копируются в первого ребенка, а остальные позиции копируются из второго родителя в упорядоченном виде слева направо, но исключаются элементы, уже попавшие в первого ребенка [10].

Операторы вставки и удаления. Данные операторы являются модификацией оператора мутации.

Оператор удаления позволяет удалять строительные блоки из родительских хромосом, тем самым создавая хромосому потомка.

Оператор вставки позволяет вставлять строительные блоки в родительскую хромосому для создания потомка.

Данные операторы эффективны для использования в ГА. Применяя оператор удаления, случайный ген со значением 0 удаляется из разбиения, тем самым расширяет некоторый кластер. Далее происходит выполнение оператора вставки, который случайным образом выбирает точку разреза. На месте точки разреза оператор фиксирует разбиение хромосомы геном со значением 0, образуя новые кластеры. Эффективность данных операторов показывает их умение быстро анализировать текущую популяцию альтернативных решений [10].

Оператор селекции. Это процесс, посредством которого хромосомы (альтернативные решения), имеющие более высокое значение ЦФ (с лучшими признаками), получают большую возможность для воспроизводства (репродукции) потомков, чем худшие хромосомы. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков. Различают элитную, один ко многим, многие к одним, лучшие с худшими, на основе методов ветвей и границ и т.д. селекции [10]. В данной работе рассматривается селекция лучшие с худшими. При кластеризации есть остаточный эффект от неправильного попадания в кластеры. Данный оператор позволяет стабилизировать состояние кластеров.

Основные этапы ГА (рис. 3).

1. Случайным образом генерируется начальная популяция. На построение популяции оказывает влияние внешняя среда в виде ЛПР. Формиру-

ется таблица характеристик для каждого гена в хромосоме.

2. Для каждой хромосомы из популяции вычисляется значение ЦФ.

3. Выполняется элитная селекция – для применения оператора кроссинговера выбираются лучшие решения.

4. Применяются генетические операторы (ГО). Первым выполняется оператор кроссинговера (ОК) – получаем потомство для последующего применения оператора мутации (ОМ).

5. К выбранным хромосомам применяется оператор мутации. При этом мутация хромосомы выполняется с некоторой вероятностью мутации, а точки мутации выбираются случайным образом.

6. Вычисляется значение ЦФ для полученных после применения оператора мутации хромосом. Таким образом, после реализации оператора мутации получается новое подмножество решений. Оно

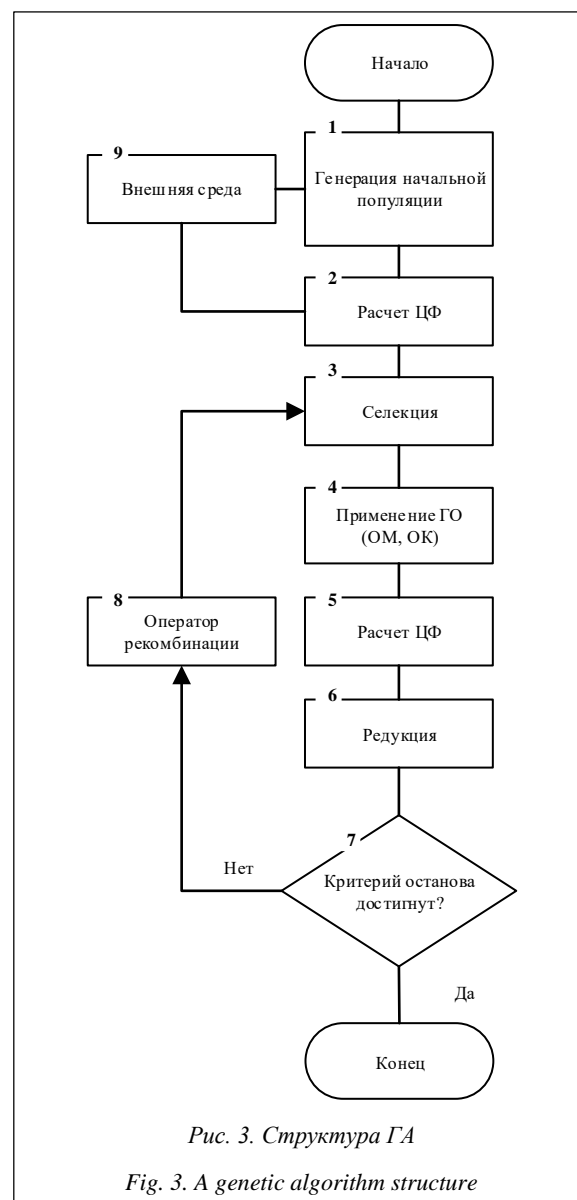


Рис. 3. Структура ГА

Fig. 3. A genetic algorithm structure

объединяется с первоначальным подмножеством решений.

7. Выполняется элитная редукция, то есть лучшие решения отбираются в следующую популяцию, а плохие удаляются. Таким образом, размер популяции остается постоянным.

8. Проверяется, достигнут ли критерий останова. В данной работе рассматривается критерий останова по времени работы алгоритма.

9. Если данный критерий не достигнут, процесс повторяется итерационно. Он выполняется на основе блока рекомбинации, который анализирует текущую популяцию альтернативных решений и управляет процессом поиска. Иначе выводятся полученные квазиоптимальные решения.

10. Конец работы алгоритма.

Пчелиный алгоритм

Оптимизационный алгоритм, моделирующий поведение пчел в живой природе, называется пчелиным алгоритмом [12, 13, 19, 20].

На начальном этапе случайным образом отправляются агенты, пытающиеся отыскать участки, где есть квазиоптимальное решение. После возвращения агенты особым образом сообщают остальным, где и сколько решений они нашли. Далее в окрестности найденных решений отправляются другие агенты, при этом, чем больше найденных решений, тем больше агентов направляются в данную окрестность. Это означает, что количество агентов в каждом направлении пропорционально значению ЦФ. Агенты продолжают искать другие квазиоптимальные решения, и процесс повторяется. Агенты живут не в двухмерной плоскости, где, зная две координаты, можно определить количество решений, а в многомерном пространстве, где один параметр функции представляется каждой координатой, которую необходимо оптимизировать. Найденное квазиоптимальное решение представляет собой значение ЦФ в этой точке.

Основные этапы работы пчелиного алгоритма (рис. 4).

1. Генерируется начальное разбиение на кластеры для поиска. Внешняя среда оказывает влияние на построение кластеров для поиска.

2. Определяется значение ЦФ для каждого кластера.

3. Выбираются начальные кластеры, на которых будет осуществляться поиск.

4. Назначается количество агентов для поиска в окрестностях кластера. При этом количество агентов обратно пропорционально значению ЦФ. То есть, чем лучше участок, тем больше пчел для поиска, чем хуже – тем меньше пчел.

5. Производится поиск на каждом кластере. Если на кластере найдено лучшее решение, оно становится кластером для поиска.

6. Выполняется случайный поиск путем генерации новых решений.

7. Проверяется критерий останова: если достигнуто заданное время работы алгоритма, выводится квазиоптимальное решение, иначе продолжается поиск в окрестностях п. 5.

8. Конец работы алгоритма.

В пчелином алгоритме, как и в генетическом, решение представлено в виде закодированного вектора. Расположение источника нектара определяется этим решением, то есть решение является координатами источника. Количество нектара определяется значением ЦФ. Так как для задачи кластеризации надо минимизировать значение ЦФ, количество нектара обратно пропорционально ЦФ. Кластер имеет начальный размер, фиксируемый параметром α .

К основным настройкам относятся количество агентов, количество источников нектара (случайно количество кластеров), размер кластеров для поиска и время работы как критерий останова.

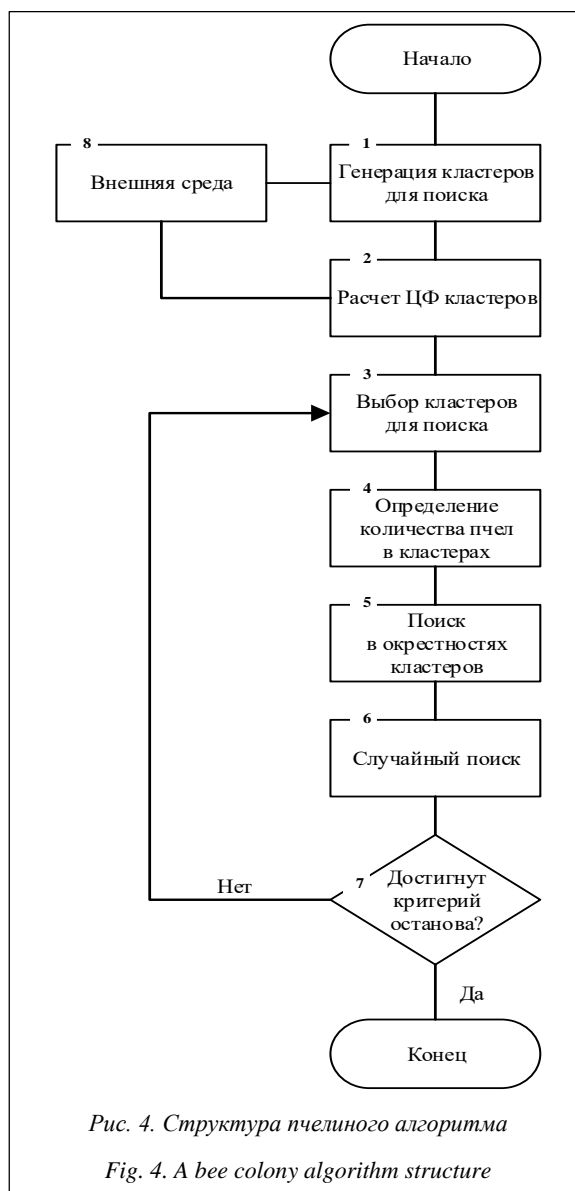


Рис. 4. Структура пчелиного алгоритма

Fig. 4. A bee colony algorithm structure

Муравьиный алгоритм

Метод роевого интеллекта описывает совместное поведение перераспределения самоформируемой системы, которая состоит из множества агентов, естественно взаимодействующих между собой и с внешней средой. Агенты просты, но, непосредственно взаимодействуя, вместе создают так называемый роевой интеллект [12, 13, 17–20].

Идея муравьиного алгоритма заключается в моделировании поведения агентов и связана с их способностью быстро находить кратчайший путь и адаптироваться к изменяющимся условиям внешней среды, находя новый кратчайший путь. При своем движении агенты пометают путь, и эта информация используется другими агентами для выбора пути. Такое элементарное правило поведения и определяет способность агентов находить новый путь, если старый оказывается недоступным [3, 12, 19].

При решении оптимизационных задач необходимо разрабатывать или модифицировать нестандартные методы решения. Агенты движутся в пространстве по случайному маршруту. Маршрут разбивается на равнозначные участки. Передвижение агентов фиксируется весовым коэффициентом, который увеличивается от прохождения агента по участку маршрута. Далее необходимо выбрать эвристику строительства решения и, если эффективное решение не найдено, реализовать оптимальный локальный поиск.

Для определения начального расположения агентов используется стратегия «Дробовик» – случайное распределение агентов в центрах кластеров, причем необязательно, чтобы численности агентов и центров совпадали.

Любой агент получает информацию о весовом коэффициенте, повышающем вероятность прохождения агента по данному участку маршрута. Весовой коэффициент в момент времени t на участке D_{ij} будет соответствовать $\tau_{ij}(t)$. Начальное значение коэффициента задается как некоторое ненулевое натуральное число ($0 \leq \tau_{ij}(t) \leq 1$). Если участок маршрута не проходит некоторое время t , то весовой коэффициент изменяется по формуле

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij,k}(t), \quad (5)$$

где m – количество агентов; p – коэффициент изменения ($0 \leq p \leq 1$).

Задача кластеризации – минимизировать среднюю сумму внутреннего кластерного расстояния:

$$\min(\sum F_i), \quad (6)$$

где F_i – длина вектора, попавшего в замкнутый участок.

Рассмотрим предложенный авторами алгоритм (рис. 5).

1. Ввод исходных данных – генерация элементов банка данных, ввод количества элементов.

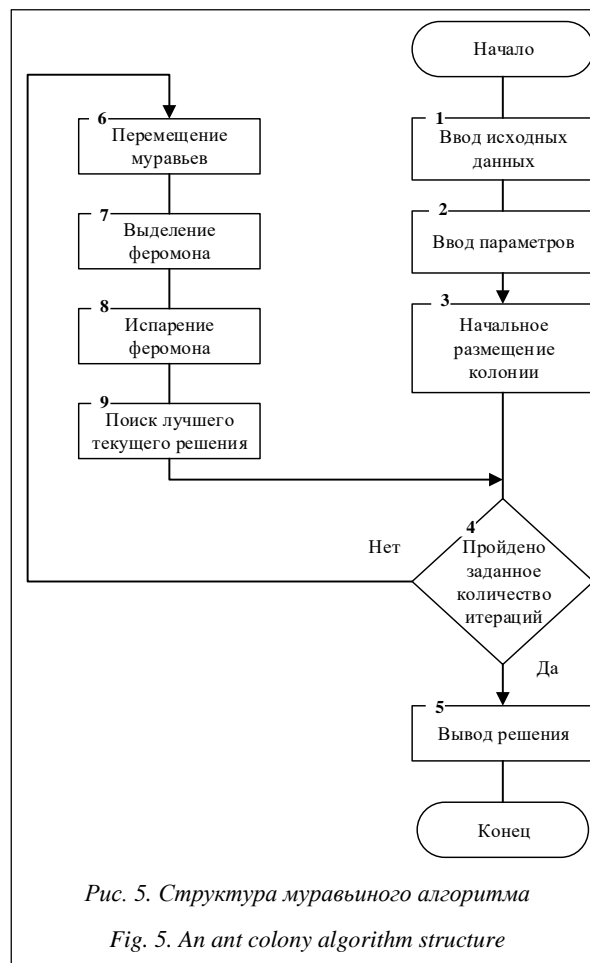


Рис. 5. Структура муравьиного алгоритма

Fig. 5. An ant colony algorithm structure

2. Ввод параметров – ввод времени работы алгоритма и количества агентов.

3. Начальное размещение агентов – случайное распределение агентов в центрах кластеров (производится с использованием стратегии «Дробовик»).

4. Осуществление итераций, количество которых определяется пользователем в п. 2.

5. Перемещение агентов – каждый агент посещает каждый элемент по одному разу.

6. Присвоение весового коэффициента участку маршрута: начальное значение весового коэффициента задается как некоторое ненулевое натуральное число ($0 \leq \tau_{ij}(t) \leq 1$).

7. Изменение весового коэффициента по формуле (5).

8. Поиск лучшего текущего решения на основании формулы (6).

Серия вычислительных экспериментов показала, что эффективность муравьиного алгоритма возрастает с увеличением размера оптимизационных задач. Важной особенностью муравьиного алгоритма является конвергенция альтернативных решений. После большого числа итераций при одновременном использовании большого количества решений нет больших задержек в локальных экстремумах. К настройкам муравьиного алгоритма относят количество агентов, весовые коэффици-

енты α , β , формирующие представление о приоритете выбранного участка маршрута. Коэффициент изменения показывает степень снижения приоритетности выбранного участка маршрута.

Разработка программного комплекса

Для эффективного решения задач интеллектуального анализа данных, в частности задачи кластеризации, предлагаются эвристические алгоритмы, преобразующие одно конечное множество альтернативных решений в другое, используя для этого механизмы и принципы генетики и эволюции живой природы [5, 9–13, 16].

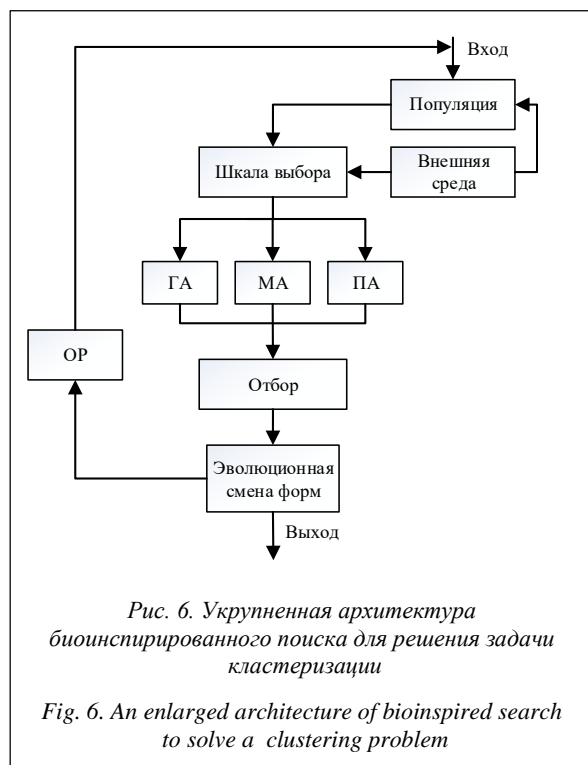
На рисунке 6 представлена укрупненная архитектура модели биоинспирированного поиска, разработанная в процессе написания программного комплекса. Данная модель анализирует поведение ЦФ при использовании трех различных эвристических алгоритмов. Использование модели дает возможность решать задачи кластеризации с любыми наборами исходных данных.

1. На входе генерируется начальная популяция альтернативных решений под воздействием внешней среды (ЛПР).

2. Выбирается алгоритм кластеризации элементов (генетический алгоритм – ГА, муравьиный алгоритм – МА; пчелиный алгоритм – ПА). На основе выполнения алгоритмов строится популяция альтернативных решений.

3. Применив отбор, ЛПР получает набор квазиоптимальных решений кластеризации.

4. Если квазиоптимальное решение не удовлетворяет, повторяем шаг 2.



5. Иначе на выходе получаются набор квазиоптимальных решений и функция времени, которая показывает эффективность представленной модели.

Основная идея данной модели – моделирование распараллеливания процесса обработки больших объемов данных с помощью эвристических алгоритмов.

Вычислительный эксперимент

Разработанные алгоритмы относятся к классу эвристических алгоритмов. Для оценки их эффективности, а именно временной сложности, производится анализ статистических данных, которые получаются при неоднократном выполнении данных алгоритмов [21–23].

Для проведения вычислительного эксперимента кластеризации были выбраны БД со 100, 200, 300, 400, 500 элементами. Перед началом эксперимента проводится настройка алгоритмов. Приведем пользовательские настройки алгоритмов.

Генетический алгоритм:

оператор кроссинговера.....упорядочивающийся
отбор.....элитный
оператор селекции.....равновероятный
вероятность оператора
кроссинговера.....10
вероятность оператора
мутации90
размер популяции.....200
время работы10 с.

Муравьиный алгоритм:

количество агентов100
вес следа феромона α 1
вес следа феромона β 3
коэффициент изменения.....0.5
время работы10 с.

Муравьиный алгоритм имеет ряд специфических настроек. Определяется, сколько агентов будут участвовать в поиске квазиоптимальных решений. Весовые коэффициенты α и β – параметры, влияющие на приоритет выбора того или иного участка маршрута. Параметр α контролирует выбор на текущем шаге i, j . Параметр β контролирует выбор на участке i, j с оптимальным значением весового коэффициента. Коэффициент изменения определяет время, с которым происходит изменение весового коэффициента на предыдущем участке.

Пчелиный алгоритм:

количество агентов a 100
количество центров кластеров.....20
количество агентов b 20
размер кластеров4
время работы10 с.

Для работы пчелиного алгоритма необходимо определиться с минимальным набором исходных

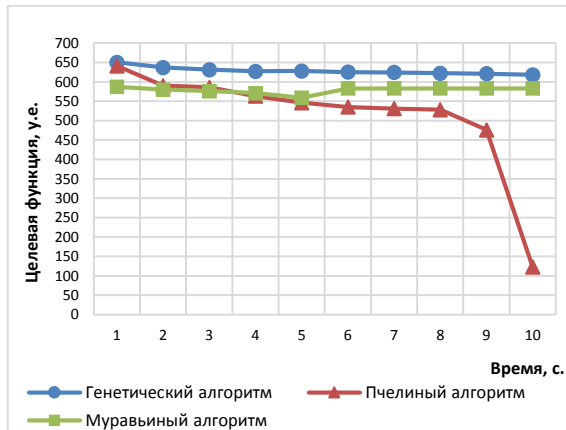


Рис. 7. Первая серия тестов

Fig. 7. The first test series

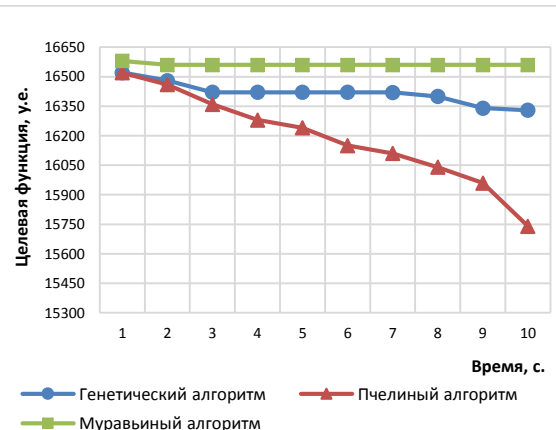


Рис. 8. Пятая серия тестов

Fig. 8. The fifth test series

данных. Количество агентов A – агентов, которые отправляются в центры кластеров для поиска эффективных решений. Если решение найдено, в него отправляются агенты для поиска других квазиоптимальных решений в окрестностях найденных решений. Количество решений является начальным назначением для агентов A . Размер кластеров измеряется в условных единицах.

Для исследования результатов эффективности в программном комплексе реализован блок вывода графика результатов выполнения, а в главном окне программы выводится квазиоптимальное значение ЦФ.

Пока не достигнут критерий останова алгоритмов на равнозначных временных участках на основании (3) минимизируется среднее внутреннее кластерное расстояние. Ось абсцисс отображает временные участки, а ось ординат – значение ЦФ на каждом участке.

Первая серия тестов: 100 элементов банка данных (рис. 7).

Значение лучших ЦФ: 1) ГА – 618, 2) МА – 583, 3) ПА – 123.

В рамках данной работы ЦФ измеряется в условных единицах для обозначения расстояния между кластерами. Из графика видно, что пчелиный алгоритм привел к наилучшему результату ЦФ за 9,36 сек. Это значение является оптимальным в рамках данной работы. Муравьиный алгоритм быстро попал в локальный оптимум. ГА показал высокую точность в поиске квазиоптимального разбиения на кластеры, даже уступая другим алгоритмам по времени поиска.

Вторая серия тестов в 200 элементов показала преимущество ГА и муравьиного алгоритма в решении поставленной задачи.

Третья серия тестов в 300 элементов показала, что пчелиный алгоритм самый быстрый и точный.

Для точности вычислительных экспериментов опустим 4-ю серию тестов на 400 элементов. Про-

ведем пятую, заключительную, серию для анализа полученных данных.

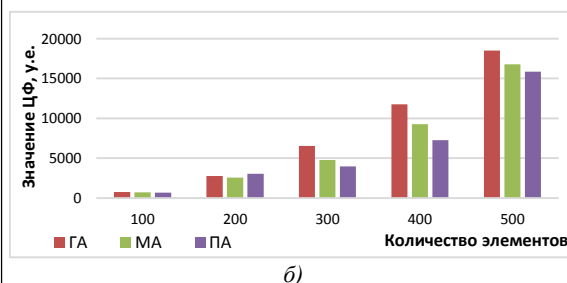
Пятая серия тестов: 500 элементов (рис. 8).

Рассмотрим вычислительный эксперимент подробно. Из графиков видно, что пчелиный и муравьиный алгоритмы практически не уступают друг другу. ГА в начале своей работы развивает хорошую скорость, но попадает в локальный оптимум на задачах с большим набором исходных данных. Данные эксперименты подтверждают способность пчелиного и муравьиного алгоритмов находить квазиоптимальное решение на задачах с большим набором исходных данных [17–20].

Таким образом, проведя пять серий экспериментов, можно сделать вывод о преимуществах каждого из алгоритмов. Использование данных алгоритмов кластеризации в задачах интеллектуаль-

Количество элементов	ГА (у.е.)	МА (у.е.)	ПА (у.е.)
100	618	583	123
200	2786	2583	3069
300	6549	4783	3962
400	11762	9265	7266
500	16311	16549	15740

а)



б)

Рис. 9. Зависимости ЦФ от количества элементов:
а) полученные результаты,
б) диаграмма зависимостей

Fig. 9. Target function dependencies on the quantity of elements a) results, б) dependency graph

ного анализа данных зависит от поставленной задачи и исходных данных. Диаграмма на рисунке 9 показывает зависимость ЦФ от количества элементов при кластеризации.

Из диаграммы видно, насколько эффективен тот или иной алгоритм. Время выполнения каждого из алгоритмов $\approx 10\,000$ мс. Эффективность ГА на большом наборе данных ниже, чем у пчелиного и муравьиного алгоритмов, но дает разбиение на кластеры, приближенное к оптимальному. Пчелиный алгоритм имеет ярко выраженное преимущество, что позволяет говорить о более точном разбиении. Муравьиный алгоритм показал не намного хуже пчелиного результат – на 7,5 сек.

В результате выполнения алгоритмов получено эффективное формирование кластеров с наименьшими ЦФ (рис. 9). Эффективность разработанных методов исследована на основе проведения серии экспериментальных вычислений.

Заключение

Разработанные алгоритмы имеют ряд преимуществ. Скорость и качество работы благоприятно сказываются на дальнейшем использовании данных алгоритмов для решения задачи кластеризации. Проведенный вычислительный эксперимент показал хорошие результаты каждого из алгоритмов. В ходе серии вычислительных экспериментов время выполнения кластеризации элементов отличалось в среднем на 3–5 % от оптимального. Под оптимальным временем поиска эффективного решения в рамках данной работы понимается значение в промежутке 8–9 секунд, полученное в результате выполнения уже разработанных алгоритмов кластеризации, например k-средних. Временная сложность разработанных алгоритмов в составе программного комплекса приблизительно равна $O(n^2)$.

Особенность данного программного комплекса – в разработанных модифицированных алгоритмах. Основной модификацией ГА является применение решета Эратосфена для начальной генерации набора альтернативных решений. Совместное применение операторов удаления и вставки дает большое преимущество в поиске квазиоптимальных решений. Модификация пчелиного и муравьиного алгоритмов заключается в модификации математического представления стандартных алгоритмов.

С учетом увеличения объемов банков данных скорость обработки уменьшается прямо пропорционально количеству элементов, что приводит к труднорешаемым проблемам. Методы и подходы эволюционного моделирования являются эффективным способом решения NP-полных, трудных задач, а модификация и разработка таких методов – актуальным направлением в интеллектуальном анализе данных.

Объединение алгоритмов позволяет улучшить качество работы программного комплекса и тем самым ускорить процесс поиска квазиоптимальных решений.

Работа выполнена при поддержке РФФИ, грант № 15-07-05523.

Литература

1. Чубукова И.А. Data Mining. М.: ИНТУИТ–БИНОМ. Лаборатория знаний, 2006. 382 с.
2. Котов А., Красильников Н. Кластеризация данных. 2006. URL: <http://www.myshared.ru/slide/177655> (дата обращения: 28.10.2016).
3. Jain A., Murty M., Flynn P. Data clustering: a review. ACM Computing Surveys, 1999, vol. 31, no. 3, pp. 264–323.
4. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. 2008. URL: <http://www.machinelearning.ru> (дата обращения: 28.10.2016).
5. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. М.: Изд-во МГУ, 2007. 14 с.
6. Witten I.H., Eibe F., Hall M.A. Data Mining: practical machine learning tools and techniques. 3rd ed. US. Elsevier Inc, 2011, 664 p.
7. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии. М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. 304 с.
8. Курейчик В.М., Полковникова Н.А. Об интеллектуальном анализе баз данных для экспертной системы // Информатика, вычислительная техника и инженерное образование. 2013. Вып. 2. С. 39–50.
9. Курейчик В.В., Родзин С.И. О правилах представления решений в эволюционных алгоритмах // Изв. ЮФУ (Технич. науки). 2010. Вып. 7. С. 13–21.
10. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: Физматлит, 2010. 306 с.
11. Курейчик В.М. Биоинспирированный поиск с использованием сценарного подхода // Изв. ЮФУ (Технич. науки). 2010. Вып. 7. С. 7–13.
12. Григораш А.С., Курейчик В.М., Курейчик В.В. [и др.]. Биоинспирированный подход к решению задач интеллектуального анализа данных. Таганрог: Изд-во ЮФУ, 2015. 100 с.
13. Кажаров А.А., Курейчик В.М. Биоинспирированные алгоритмы. Решение оптимизационных задач. Germany: LAP Lambert Acad. Publ., 2011. 73 p.
14. Кравченко Ю.А., Запорожец Д.Ю., Лежебоков А.А. Способы интеллектуального анализа данных в сложных системах // Изв. КБНЦ РАН. 2012. Вып. 3. С. 52–57.
15. Tso B., Mather P.M. Classification methods for remotely sensed data. US, CRC Press, 2009, 349 p.
16. Курейчик В.В., Курейчик В.М. Генетический алгоритм определения пути коммивояжера // Изв. РАН (Теория и системы управления). 2006. Вып. 1. С. 94–100.
17. Редько В.Г. Моделирование когнитивной эволюции. На пути к теории эволюционного происхождения мышления. М.: Изд-во ЛЕНАНД, 2015. 256 с.
18. Сегаран Т. Программируемый коллективный разум; [пер. с англ.]. СПб: Символ-Плюс, 2015. 368 с.
19. Курейчик В.М., Кажаров А.А. Использование роевого интеллекта в решении NP-трудных задач // Изв. ЮФУ (Технические науки). 2011. Вып. 7. С. 30–36.
20. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes Univ. Publ., 2005, 10 p.
21. Cormen T.H., Leiserson C.E., Rivest R.L., Stin C. Introduction to Algorithms. USA, MIT Publ., 2009 1296 p.
22. Мандель И.Д. Кластерный анализ. М.: Финансы и статистика, 1988. 342 с.
23. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.

CLUSTERING SOFTWARE SOLUTION

A.S. Grigorash¹, Postgraduate Student, grigoraschandrey@gmail.com

V.M. Kureychik¹, Dr.Sc. (Engineering), Head of the Department, kur@tgn.sfedu.ru

V.V. Kureychik¹, Dr.Sc. (Engineering), Head of the Department, vkur@sfedu.ru

¹ Southern Federal University, Nekrasovsky Lane 44, Taganrog, 347922, Russian Federation

Abstract. Nowadays, information technologies have developed considerably. Databases grow too fast, and data processing becomes more and more difficult with each passing day. Data mining methods become an urgent area of study for processing large amounts of data because they allow finding out implicit patterns in data sets. One of the main tasks of data mining is clustering. The authors formulated the clustering problem. Clustering is a NP-complete, difficult task, therefore it is necessary to develop non-standard algorithms and methods for finding an effective solution in polynomial time.

The purpose of this research is to build a software package for solving the clustering problem. In order to solve it, there were proposed modified methods of evolutionary modeling and swarm intelligence, which might adapt to environmental changes.

The paper specifies a modification to find quasi-optimal solutions, which allowed significantly reducing the time of cluster formation. It also describes modified genetic, ant and bee colony algorithms, as well as an algorithm functioning model as a consistent system. The article considers the structure of the software system and the developed an intuitive user interface.

A computational experiment was carried out on different amounts of data banks. A series of computational experiments showed that the time for element clustering approximately equals to 9.4 seconds. The difference between the obtained value and the optimal value is at average of 3–5 %. Within the research, the optimal time is 8–9 seconds. The experiment revealed temporary complexity of the developed complex. It approximately equals to the ICA $\approx O(n^2)$.

Keywords: data mining, evolutionary modeling, genetic algorithm, swarm intelligence, bee algorithm, ant algorithm.

Acknowledgements. The work has been done with support from RFBR, grant no. 15-07-05523.

References

1. Chubukova I.A. *Data Mining*. Moscow, INTUIT–BINOM: Laboratoriya znany Publ., 2006, 382 p.
2. Kotov A., Krasilnikov N. *Klasterizatsiya dannykh* [Data Clustering]. Seminar. 2006. Available at: <http://www.myshared.ru/slide/177655> (accessed October 28, 2016).
3. Jain A., Murty M., Flynn P. Data clustering: A review. *ACM Computing Surveys*. 1999, vol. 31, no. 3, pp. 264–323.
4. *Professionalny informatsionno-analitichesky resurs, posvyashchenny mashinnomu obucheniyu, raspoznavaniyu obrazov i intellektualnomu analizu dannykh* [Professional Information and Analytical Website Devoted to Machine Learning, Image Identification and Data Mining]. 2008. Available at: <http://www.machinelearning.ru> (accessed October 28, 2016).
5. Vorontsov K.V. *Algoritmy klasterizatsii i mnogomernogo shkalirovaniya* [Clustering and Multidimensional Scaling Algorithms]. Moscow, MGU Publ., 2007, 14 p.
6. Witten I.H., Eibe F., Hall M.A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. US, Elsevier Inc. Publ., 2011, 664 p.
7. Bashmakov A.I., Bashmakov I.A. *Intellektualnye informatsionnye tekhnologii* [Intelligent Information Technologies]. Moscow, N.E. Bauman MSTU Publ., 2005, 304 p.
8. Kureychik V.M., Polkovnikova H.A. About intellectual analysis of the database for expert system. *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Computer Science, Computer Engineering and Engineering Education]. 2013, iss. 2 (13), pp. 39–50 (in Russ.).
9. Kureychik V.V., Rodzin S.I. On the rules for the submission decisions in evolutionary algorithm. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2010, no. 7 (108), pp. 13–21 (in Russ.).
10. Gladkov L.A., Kureychik V.V., Kureychik V.M. *Geneticheskie algoritmy* [Genetic Algorithms]. Moscow, Fizmatlit Publ., 2010, 306 p.
11. Kureychik V.M. Bioinspired search based on scenario method. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2010, no. 7 (108), pp. 7–13 (in Russ.).
12. Grigorash A.S., Kureychik V.M., Kureychik V.V. *Bioinspirovanny podkhod k resheniyu zadach intellektualnogo analiza dannykh* [Bioinspired Approach to Solving Data Mining Problems]. Monograph. Taganrog, SFedU Publ., 2015, 100 p.
13. Kazharov A.A., Kureychik V.M. *Bioinspirovannyye algoritmy. Reshenie optimizatsionnykh zadach* [Bioinspired Algorithms. Solving Optimization Problems]. Germany, LAP LAMBERT Academic Publ., 2011, 73 p.
14. Kravchenko Yu.A., Zaporozhets D.Yu., Lezhebokov A.A. Methods of intelligent data analysis in complex systems. *Izvestiya KBNTs RAN* [News Kabardin-Balkar Scientific Center of Russian Academy Sciences]. 2012, no. 3 (47), pp. 52–57 (in Russ.).
15. Tso B., Mather P.M. *Classification Methods for Remotely Sensed Data*. US, CRC Press, 2009, 339 p.
16. Kureychik V.V., Kureychik V.M. A genetic algorithm for finding a salesman's route. *Jour. of Computer and Systems Sciences Int.* 2006, no. 1, pp. 94–100 (in Russ.).
17. Redko V.G. *Modelirovaniye kognitivnoy evolyutsii. Na puti k teorii evolyutsionnogo proiskhozhdeniya myshleniya* [Modeling Cognitive Evolution: on the Way to the Theory of Ideation Evolutionary Descent]. Moscow, LENAND Publ., 2015, 256 p.
18. Segaran T. *Programming Collective Intelligence*. O'Reilly Media, 2007, 362 p. (Russ.ed.: St. Petersburg, Simvol-Plyus Publ., 2015, 368 p.).
19. Kureychik V.M., Kazharov A.A. Swarm intelligence using for np-tasks solving. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2011, no. 7 (120), pp. 30–36 (in Russ.).
20. Karaboga D. *An idea based on honey bee swarm for numerical optimization*. Technical Report TR06, Erciyes Univ. Publ., 2005, 10 p.
21. Cormen T.H., Leiserson Ch.E., Rivest R.L., Stein C. *Introduction to Algorithms*. US, MIT Publ., 2009, 1296 p.
22. Mandel I. D. *Klasterny analiz* [Cluster Analysis]. Moscow, Finansy i statistika Publ., 1988, 342 p.
23. Ayvazyan S.A., Bukhshtaber V.M., Enyukov I.S., Meshalkin L.D. *Prikladnaya statistika: klassifikatsiya i snizhenie razmernosti* [Applied Statistics: Classification and Dimension Reduction]. Moscow, Finansy i Statistika Publ., 1989, 607 p.

УДК 519.688

DOI: 10.15827/0236-235X.118.270-281

Дата подачи статьи: 10.10.16

2017. Т. 30. № 2. С. 270–281

РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ НЬЮТОНОВСКИХ МЕТОДОВ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

А.Б. Свириденко, преподаватель, roshechka@gmail.com
(Новороссийский филиал Кубанского государственного университета,
ул. Героев Десантников, 87, г. Новороссийск, 353900, Россия)

Предложен подход к увеличению эффективности ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных. В основе увеличения эффективности ньютоновских методов лежит модифицированное разложение Холецкого матрицы вторых производных, определяющее решение проблемы масштабирования шагов при спуске, аппроксимацию неквадратичными функциями, интеграцию с методом доверительной окрестности и уменьшение нормы априорной поправки. Исследована возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой.

Рассмотрена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц и предлагаемого подхода к увеличению эффективности численных методов ньютоновского типа – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой.

Приведены описания программных реализаций, все версии алгоритмов реализованы на языке Visual Basic .NET, среда разработки – Microsoft Visual Studio 2010. Приведены результаты численного исследования эффективности реализованных алгоритмов с учетом ряда правил, описанных в работе.

Изучен подход к увеличению эффективности ньютоновских методов с конечно-разностной аппроксимацией первых и вторых производных. Подход является основой для дальнейших исследований, результаты которых могут быть использованы для построения численных методов ньютоновского типа.

Ключевые слова: ньютоновские методы, структура матрицы, конечно-разностная аппроксимация, факторизация Холецкого, масштабирование шагов, метод доверительной окрестности, норма априорной поправки, бенчмаркинг, профиль производительности, коллекция тестовых задач.

Ньютоновские методы – фундаментальные инструменты численного анализа, исследования операций оптимизации и управления. Например, большинство наиболее известных методов в линейном и нелинейном программировании строятся на их основе. Общий принцип построения большинства ньютоновских методов опишем следующим образом.

Рассмотрим задачу безусловной минимизации

$$\min_{x \in R^n} F(x), \quad (1)$$

где $F: R^n \rightarrow R^1$ – гладкая функция; R^n – n -мерное евклидово пространство.

Пусть h^k, H^k – градиент и гессиан, вычисленные на итерации k в точке x^k процесса решения задачи (1). Тогда общий принцип построения большинства ньютоновских методов безусловной оптимизации с регулировкой шага состоит в следующем. На каждой итерации сначала строится некоторая связанная с H^k существенно положительно определенная матрица \bar{H}^k , а затем направление спуска p^k вычисляется как решение системы $\bar{H}^k p^k = -h^k$ и определяется новая точка $x^{k+1} = x^k + \alpha_k p^k$, где α_k – длина шага, для которого $F^{k+1} < F^k$. При этом процедуру построения \bar{H}^k организуют так, чтобы \bar{H}^k совпадала с исходной матрицей H^k , если последняя сама является положительно определенной, причем выяснение определенности H^k и построение

\bar{H}^k осуществляются параллельно в рамках одной процедуры на основе некоторых матричных разложений, которые позволяют выявить знаки собственных чисел H^k и приспособиться для генерации \bar{H}^k .

На практике условие гладкости целевой функции является излишне строгим: как правило, или первые и (или) вторые производные не существуют, или процесс вычисления этих производных слишком трудоемок. В этом случае быстро сходящийся алгоритм можно построить на основе их конечно-разностной аппроксимации. Подобно своему классическому прототипу он порождает целый класс модифицированных методов, причем модификации, предлагаемые для обобщения метода Ньютона на случаи со знаконеопределенными H^k , применимы и к его конечно-разностному аналогу, и наоборот [1, 2].

В данной работе в основе выяснения определенности H^k и построения \bar{H}^k лежит факторизация (разложение) Холецкого. Разложение Холецкого для симметричной положительно определенной матрицы H^k имеет вид $H^k = L^k D^k (L^k)^T = (U^k)^T D^k U^k$, где $U^k = (L^k)^T$ – верхняя треугольная матрица с диагональными элементами, равными единице; D^k – диагональная матрица с положительными ведущими элементами по диагонали. Поскольку столбцы матрицы L^k с номерами с 1-го по $(j-1)$ -й

известны, ее j -й столбец определяется по формулам

$$d_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} d_{ss}^k (l_{js}^k)^2, \quad l_{ij}^k = \frac{1}{d_{ii}^k} \left(h_{ij}^k - \sum_{s=1}^{i-1} d_{ss}^k l_{js}^k u_{is}^k \right).$$

Аналогичные формулы существуют и для построчной организации вычислений. В отличие от гауссовых исключений алгоритм Холецкого численно устойчив без каких-либо перестановок [3]. Это свойство определяется соотношением

$$\sum_{i=1}^n (l_{ij}^k)^2 d_{jj}^k = h_{jj}^k, \quad j = 1, \dots, n, \text{ между элементами } H^k \text{ и } L^k.$$

В силу него существует априорное ограничение сверху на элементы $r_{ij}^k = l_{ij}^k \sqrt{d_{jj}^k}$: каждый из них не превосходит максимальную величину h_{jj}^k . Соответственно лавинообразный рост элементов r_{ij}^k невозможен независимо от того, будут ведущие элементы малыми или нет.

Построение направлений ньютоновского поиска. Факторизация Холецкого неприменима для любой симметричной матрицы по следующим причинам. Во-первых, для знаконеопределенной матрицы H^k факторизация Холецкого может не существовать. Во-вторых, даже если она и существует, то гарантировать численную устойчивость алгоритма уже нельзя, поскольку никаких априорных ограничений на субдиагональные элементы L^k в рассматриваемом случае не будет. Итак, на основе обычной факторизации Холецкого построение \bar{H}^k не получить, для этого нужно воспользоваться модифицированной факторизацией. Наиболее эффективная вычислительная схема предложена в [2, 3]. Данный подход состоит в том, чтобы строить факторы Холецкого D^k и L^k , подчиняющиеся двум требованиям: все диагональные элементы D^k должны быть существенно положительными, модули всех элементов треугольного фактора L^k должны быть равномерно ограничены сверху. Точнее говоря, требуется, чтобы для всех $j=1, 2, \dots, n$ и некоторых заданных положительных δ и β выполнялись неравенства

$$d_{jj}^k \geq \delta, \quad |r_{ij}^k| \leq \beta, \quad i > j, \quad (2)$$

где r_{ij}^k – введенные для удобства изложения вспомогательные величины, по определению равные $u_{ij}^k \sqrt{d_{jj}^k}$. Гринштадт предложил в [4] при реализации алгоритма на конкретной ЭВМ, в которой под запись мантииссы отводится t битов, величину δ вычислять по формуле $\delta = \max \{2^{-t} \|H^k\|_{\infty}, 2^{-t}\}$.

Для сохранения численной устойчивости процедуры построения H^k , а также для совпадения H^k и \bar{H}^k в случае положительно определенной H^k целесообразно величину β вычислять по формуле $\beta^2 = \max \left\{ \gamma, \xi / \sqrt{n^2 - 1}, \varepsilon_M \right\}$.

Здесь ξ – максимальный модуль недиагонального элемента H^k ; γ – значение максимального из диагональных элементов H^k ; ε_M – машинная точность; ε_M вводится в формулу расчета β , чтобы обеспечить устойчивость вычислений, когда норма H^k очень мала [4]. При этом процедура расчета модифицированных факторов L^k и D^k фактически представляет собой обычный алгоритм факторизации Холецкого с попутным увеличением (по мере необходимости) диагонали исходной матрицы с целью добиться выполнения неравенств (2). Матрицы L^k и D^k , полученные по окончании описанной процедуры, будут факторами Холецкого для положительно определенной матрицы \bar{H}^k , связанной с H^k следующим образом: $L^k D^k (L^k)^T = H^k + E^k + \bar{H}^k$, где E^k – неотрицательная диагональная матрица, j -й элемент которой равен e_{jj}^k . Таким образом, положительно определенная матрица \bar{H}^k может отличаться от исходной матрицы H^k только диагональными элементами.

Направление спуска p^k вычисляется как решение системы $\bar{H}^k p^k = -h^k$, причем $p^k = -(\bar{H}^k)^{-1} h^k$

определяют последовательным решением двух систем линейных уравнений с треугольными матрицами: $L^k y^k = -h^k$, $D^k (L^k)^T p^k = y^k$.

Для построения модифицированного разложения Холецкого требуется выполнить около $\frac{1}{6} n^3$ арифметических операций, примерно столько же, сколько необходимо для обычного разложения для положительно определенной матрицы.

Вычислительная схема 1 модифицированного разложения Холецкого. Далее приводится детальное описание всех операций, выполняемых по ходу построения модифицированного разложения Холецкого с перестановками [3]. Дана наиболее эффективная схема организации расчетов. В процессе вычисления j -го столбца матрицы L^k участвуют вспомогательные величины $c_{is}^k = l_{is}^k d_{ss}^k$, $s = 1, \dots, j$, $i = j, \dots, n$.

Шаг 0 (Расчет порога для элементов). Вычислить $\beta^2 = \max \{ \gamma^0, \xi / v, \varepsilon_M \}$, где $v = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k .

Шаг 1 (Инициализация). Присвоить индексу столбца j значение 1. Принять $c_{ii}^k = g_{ii}^k$, $i = 1, \dots, n$.

Шаг 2 (Перестановка строк и столбцов). Найти индекс q , такой, что $|c_{qq}^k| = \max_{j \leq i \leq n} |c_{ii}^k|$. Поменять местами все данные, отвечающие столбцам матрицы H^k с номерами q и j , а затем проделать то же самое с данными, отвечающими ее q -й и j -й строкам.

Шаг 3 (Поиск максимальной по модулю величины $l_{ij}^k d_{ii}^k$). Вычислить $c_{ij}^k = g_{ij}^k - \sum_{s=1}^{j-1} l_{js}^k c_{is}^k$ для

$i = j+1, \dots, n$ и найти $\theta_j = \max_{j+1 \leq i \leq n} |c_{ij}^k|$ (если $j = n$, взять $\theta_j = 0$).

Шаг 4 (Расчет j -го диагонального элемента фактора D^k). Вычислить $d_{jj}^k = \max \{ \delta, |c_{jj}^k|, \theta_j^2 / \beta^2 \}$ и поправку $e_{jj}^k = d_{jj}^k - c_{jj}^k$. Если $j = n$, вычисления прекратить.

Шаг 5 (Расчет j -й строки L^k). Вычислить $l_{js}^k = c_{js}^k / d_{ss}^k$ для $s = 1, \dots, j-1$. Для $i = j+1, \dots, n$ пересчитать диагональные элементы c_{ii}^k по формуле $c_{ii}^k = c_{ii}^k - l_{ij}^k c_{ij}^k$. Присвоить j значение $j+1$ и вернуться к шагу 2.

Модифицированная факторизация Холецкого, помимо прочего, позволяет определить и направление отрицательной кривизны, решая уравнение $(L^k)^T p^k = e_s$, где индекс s выбирается из условия $d_{ss}^k - e_{ss}^k \leq d_{jj}^k - e_{jj}^k$, $j = 1, \dots, n$.

Подход к увеличению эффективности алгоритма Гилла и Мюррея предложен в [1]. Доказано, что подход к построению направления спуска p^k определяет решение проблемы масштабирования шагов при спуске и интеграцию с методом доверительной окрестности следующим образом. В алгоритме Гилла и Мюррея направление спуска p^k в конечном счете вычисляется как решение системы

$$(L^k)^T p^k = U^k p^k = u^k, \quad (3)$$

где $u^k = (D^k)^{-1} y^k$, а y^k – это решение системы $L^k y^k = -h^k$. Величина элементов $|u_j^k|$ зависит от способа задания направления спуска. Следуя подходу Гилла и Мюррея, для численной устойчивости расчета элементов D^k , L^k , u^k достаточно потребовать изменения способа задания p^k так, чтобы для всех $j = 1, 2, \dots, n$ и некоторых заданных положительных δ , β и ω выполнялись неравенства

$$d_{jj}^k \geq \delta, |r_{ij}^k| \leq \beta, i > j, |u_j^k| \leq \omega.$$

Это может стать ключом к решению проблемы масштабирования шагов при спуске, но такое предположение ошибочно: подход линейной алгебры к вычислению направления спуска p^k исключает расчет элементов вектора u^k по ходу построения модифицированного разложения Холецкого [5, 6]. Нужен альтернативный подход, который опишем следующим образом.

Равенства $H^k = (U^k)^T D^k U^k$ достаточно для определения элементов матриц D^k , U^k . В скалярной форме это равенство выглядит следующим образом: $h_{ij}^k = \sum_{\mu=1}^i u_{\mu i}^k u_{\mu j}^k d_{\mu\mu}^k$. Отсюда, полагая $u_{ii}^k = 1$, построим соотношения

$$d_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} (u_{si}^k)^2 d_{ss}^k, \quad (4)$$

$$u_{ij}^k = \left(h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k \right) / d_{ii}^k, j > i,$$

для расчета элементов D^k , U^k . Построение факторов Холецкого математически эквивалентно применению метода исключения Гаусса к системе уравнений $H^k p^k = -h^k$ в прямом порядке, при этом техника исключения Гаусса позволяет получить соотношение

$$u_i^k = - \left(h_i^k - \sum_{s=1}^{i-1} u_s^k u_{si}^k d_{ss}^k \right) / d_{ii}^k \quad (5)$$

для расчета элементов вектора u^k и систему уравнений (3) для вычисления направления спуска p^k . Интеграция техники исключения Гаусса и факторизации Холецкого приводит к соотношениям (4) и (5), которые с помощью элементарных арифметических операций позволяют вычислить элементы u^k по ходу построения факторов Холецкого.

Интеграция техники исключения Гаусса и факторизации Холецкого генерирует множество реализаций численно устойчивых вычислительных схем задания направления спуска p^k , в основе которых лежит требование, чтобы на очередном шаге вычислений коэффициентов факторов Холецкого соответствующий диагональный элемент матрицы D^k и соответствующий элемент вектора u^k сначала рассчитывались по вычисленным ранее значениям этих коэффициентов. Затем диагональный элемент D^k увеличивается настолько, насколько необходимо, чтобы все диагональные элементы D^k были существенно положительными, модули всех элементов U^k и u^k были равномерно ограничены сверху. Например, достаточно потребовать, чтобы для всех $i = 1, 2, \dots, n$ выполнялись неравенства

$$d_{ii}^k \geq \delta, |u_{ij}^k| \leq 1, j > i, |u_j^k| \leq 1, \quad (6)$$

положив $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, поскольку элементы d_{ii}^k , c_{ij}^k / d_{ii}^k и c_i^k / d_{ii}^k определяют i -е строки матриц D^k , U^k , u^k и удовлетворяют неравенствам (6). Здесь c_{ii}^k , c_{ij}^k , c_i^k – вспомогательные величины, определяемые следующим образом:

$$c_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} (u_{si}^k)^2 d_{ss}^k,$$

$$c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k, j > i, \quad (7)$$

$$c_i^k = - \left(h_i^k - \sum_{s=1}^{i-1} u_s^k u_{si}^k d_{ss}^k \right).$$

Интеграция техники исключения Гаусса и факторизации Холецкого генерирует множество реализаций вычислительных схем интеграции с методом доверительной окрестности, например, следующим образом. В процессе построения D^k , U^k , u^k вычислить элементы $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$), где $l_i^k = \max \{ \delta, |d_{ii}^k| \}$, вычислить $\gamma^k = \max_{1 \leq i \leq n} g_i^k$ и определить направление спуска p^k из системы

$$\frac{1}{\gamma^k} \bar{H}^k p^k = -h^k. \quad (8)$$

Здесь параметр γ задается пользователем, число γ^k ($\gamma^k \geq 1$ по построению) характеризует степень однородности, при этом (8) – простейшая форма аппроксимации, отличная от квадратичной. Пусть $\bar{H}^k p^k = -\gamma^k h^k = -h^k - (\gamma^k - 1)h^k = -h^k - G^k p^k$, где $G^k -$ диагональная матрица с элементами $g_{ii}^k = \frac{(\gamma^k - 1)h_i^k}{p_i^k}$

на диагонали. Тогда $(\bar{H}^k + G^k) p^k = -h^k$, что означает сдвиг на g_{ii}^k всех i -х собственных значений матрицы \bar{H}^k . В методах с регулировкой шага матрицу H^k модифицируют так, чтобы изменения не затрагивали подпространства, натянутого на ее собственные векторы с положительными собственными значениями [3]. Если же замена осуществляется в методе доверительной окрестности, она отражается на всех векторах, так как результатом замены H^k на матрицу $\bar{H}^k = H^k + \lambda^k I$ будет сдвиг на λ^k всех собственных значений матрицы H^k [3]. Из вышесказанного следует, что стратегия выбора направления спуска определяет и интеграцию с методом доверительной окрестности.

Вычислительная схема 2 модифицированного разложения Холесского. Далее приведена наиболее эффективная схема организации расчетов. Все фигурирующие в ней величины при реализации на ЭВМ могут размещаться в памяти, первоначально выделяемой для записи матриц H^k и h^k . При этом коэффициенты рассчитываемых факторов занимают места ее использованных элементов. В процессе вычисления i -й строки фактора U^k участвуют вспомогательные величины (7). В процессе построения системы уравнений $U^k p^k = u^k$ на каждой итерации k выявляются знаки собственных чисел и вычисляется количество отрицательных n_o и нулевых n_0 собственных значений, число переходов к направлению отрицательной кривизны n_S .

Шаг 0. Вычислить $\beta^2 = \max \{ \gamma^0, \xi / \nu, \varepsilon_M \}$, где $\nu = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ^0 и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k . Положить $\varepsilon_0 = 2^{-\frac{\tau_F}{2}}$, $\varepsilon_s = 2^{-\frac{\tau_F}{3}} (1 + \|F(x^{k-1}) - F(x^k)\|)$. Здесь τ_F – число правильных разрядов $F(x^k)$, которые хотелось бы получить.

Шаг 1. Принять $i = 1$ и $c_{jj}^k = h_{jj}^k$ ($j = 1, \dots, n$), $c_j^k = -h_j^k$ ($j = 1, \dots, n$).

Шаг 2. Найти такой индекс q , что $|c_{qq}^k| = \max_{i \leq j \leq n} |c_{ij}^k|$, и поменять местами все данные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Шаг 3. Если $c_{ii}^k < \varepsilon_0$, то положить $l_i^k = \delta$; иначе положить $l_i^k = c_{ii}^k$.

Шаг 4. Если $|c_{ii}^k| \leq \varepsilon_0$, то положить $n_0 = n_0 + 1$ и перейти к шагу 8; иначе перейти к шагу 5.

Шаг 5. Если $c_{ii}^k > 0$, то перейти к шагу 8; иначе перейти к шагу 6.

Шаг 6. Принять $n_0 = n_0 + 1$. Если $\|h^k\| > \varepsilon_s$, то перейти к шагу 8; иначе перейти к шагу 7.

Шаг 7. Построить систему уравнений $U^k p^k = e_i$ для поиска направления отрицательной кривизны по формулам $u_j^k = 0$, $j = 1, \dots, n$, $u_i^k = 1$, положить $n_S = n_S + 1$ и остановиться.

Шаг 8. Если $i = n$, то положить $\theta_i = 0$; иначе вычислить вспомогательные величины c_{ij}^k , $j = i + 1, \dots, n$, по формуле $c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k$ и найти

$$\theta_i = \max_{i+1 \leq j \leq n} |c_{ij}^k|.$$

Шаг 9. Вычислить диагональный элемент фактора D^k и элемент вектора u^k по формулам $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, $u_i^k = c_i^k / d_{ii}^k$. Вычислить $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$). Если $i = n$, то перейти к шагу 11.

Шаг 10. Вычислить строку $u_{ij}^k = c_{ij}^k / d_{ii}^k$, $j = i + 1, \dots, n$, фактора U^k , пересчитать вспомогательные величины по формулам $c_{jj}^k = c_{jj}^k - u_{ij}^k c_{ij}^k$, $j = i + 1, \dots, n$, $c_j^k = c_j^k - u_{ij}^k c_i^k$, $j = i + 1, \dots, n$, положить $i = i + 1$ и перейти к шагу 2.

Шаг 11. Найти $\gamma^k = \max_{1 \leq i \leq n} g_i^k$, положить $u^k = \gamma^k u^k$ и остановиться.

Замечания. Предложенный подход к вычислению длины шага можно найти в [7–9]. В основе лежит интеграция подходов [3, 10–12].

Предложенный подход к модификации критериев останова, разработанных Гиллом, Мюрреем и Райт [3], можно найти в [6].

Предложенный подход к оцениванию конечно-разностных интервалов и подход к конечно-разностной аппроксимации первых и вторых производных можно найти в [1]. В основе лежит интеграция подходов [3, 11], там же описаны способы выбора конечно-разностных интервалов. Стандартные программы численного дифференцирования нацелены на поиск конечно-разностного интервала, обеспечивающего минимальную суммарную ошибку вычисляемого приближения. Алгоритмы выбора интервала для центральной конечно-разностной формулы можно найти, например, в [13]. Однако для оптимизационных приложений такие программы не подходят, так как, обеспечивая избыточную точность вычисления

оценок производных, они обычно требуют слишком большого числа обращений к процедуре расчета значений функции.

В [14] отмечено, что замена градиентов их конечно-разностными оценками лучше всего проходит в квазиньютоновских методах. Для методов ньютоновского типа результаты получаются менее удовлетворительными.

Тестирование

Построение алгоритмов, в том числе и развитие подхода Гилла и Мюррея к построению направления спуска [1, 6, 9, 14], требует постоянного тестирования на больших классах тестовых задач различных структур и характеристик с различными настройками с целью выявления слабых мест – ситуаций, приводящих к замедлению алгоритма или ухудшению качества результата и т.д.

Алгоритмы безусловной оптимизации должны быть протестированы по крайней мере в двух различных смыслах:

- для оценки конкретной реализации и соответствующих вычислительных процедур; очевидно, что хорошо разработанные тестовые задачи являются очень мощным инструментом для уточнения концепций и механизмов алгоритма;
- для получения представления о гипотезе, доказательстве работоспособности алгоритма и сравнения алгоритмов на экспериментальном уровне.

Как правило, задачи безусловной оптимизации можно разделить на два типа: искусственные и реальные. Искусственные задачи используются для того, чтобы увидеть поведение алгоритма в различных трудных ситуациях, таких как оптимизация функций с длинными узкими оврагами, со значительными нуль-пространственными эффектами, унимодальных, с огромным числом существенных локальных экстремумов и т.д. Основной характеристикой искусственных задач является то, что их относительно легко настраивать и применять в процессе построения алгоритма.

Реальные задачи берутся из разных источников прикладных оптимизационных задач физики, химии, техники, биологии, экономики, океанологии, астрономии, метеорологии и т.д. В отличие от искусственных настройки реальных задач труднодоступны и ими нелегко оперировать. Они могут иметь сложные алгебраические или дифференциальные выражения, могут зависеть от огромного количества данных и, возможно, зависеть от некоторых параметров, которые должны быть оценены определенным образом. Очень хорошая коллекция реальных задач безусловной оптимизации описана в работах [15, 16]. Следует отметить, что собрание коллекций реальных задач началось в конце 1970-х годов с отстаивания высокого качества отчетности о вычислительных экспериментах математического программирования с ПО [17–19].

Источником тестовых задач, рассмотренных в данной работе, является коллекция [20], в состав которой вошли тестовые функции из коллекции CUTE и прочих [21–24], а также из других работ и технических отчетов. Цель этой коллекции – представить большое количество общих тестовых функций, которые могут использоваться для тестирования алгоритмов безусловной оптимизации и сравнения исследований. Для каждой функции даны алгебраические представления и начальные (стартовые) точки. Для всех задач из коллекции [20] известные алгоритмы безусловной оптимизации, например, квазиньютоновские, сходятся к одному и тому же локальному решению.

Опишем конкретные тестовые задачи [20].

Задача 1 (Extended Rosenbrock function):

$$F(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 2 (Wood function):

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1),$$

$$x^0 = (-3 \ -1 \ -3 \ -1)^T.$$

Задача 3 (Generalized Rosenbrock function):

$$F(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 4 (Extended White & Holst function):

$$F(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 5 (Extended Penalty function):

$$F(x) = \sum_{i=1}^{n-1} 100(x_i - 1)^2 + \left(\sum_{j=1}^n x_j^2 - 0.25 \right)^2,$$

$$x^0 = (1 \ 2 \ \dots \ n)^T.$$

Задача 6 (Perturbed Quadratic function):

$$F(x) = \sum_{i=1}^n ix_i^2 + \frac{1}{100} \left(\sum_{i=1}^n x_i \right)^2,$$

$$x^0 = (0.5 \ 0.5 \ \dots \ 0.5)^T.$$

Задача 7 (Raydan 1 function):

$$F(x) = \sum_{i=1}^n \frac{i}{10} (\exp(x_i) - x_i), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Задача 8 (Raydan 2 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - x_i), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Задача 9 (Diagonal 1 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - ix_i), \quad x^0 = (1/n \ 1/n \ \dots \ 1/n)^T.$$

Задача 10 (Diagonal 2 function):

$$F(x) = \sum_{i=1}^n \left(\exp(x_i) - \frac{x_i}{i} \right), \quad x^0 = (1/1 \ 1/2 \ \dots \ 1/n)^T.$$

Задача 11 (Diagonal 3 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - i \sin(x_i)), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Завершение итераций производилось с использованием рекомендаций из [6]:

$$|F(x^{k-1}) - F(x^k)| < 2^{-\tau_F} (1 + |F(x^{k-1}) - F(x^k)|),$$

$$\|h^k\| \leq 2^{-\frac{\tau_F}{3}} (1 + |F(x^{k-1}) - F(x^k)|),$$

$$\|h^k\| \leq 2^{-\frac{\tau_F}{3}} (1 + |F(x^k)|),$$

что является ужесточением условий останова из [4], позволяющим повысить точность решения, и хорошо согласуется с численными экспериментами [1].

Замечания. Трудности сравнения нескольких методов оптимизации по результатам численных экспериментов состоят в следующем.

Во-первых, сравнению подвергаются не методы, а машинные реализации соответствующих алгоритмов. Результат сравнения зависит от качества программного кода, выбора параметров алгоритма, числовых типов данных и т.д.

Во-вторых, неясно, каким образом соизмерять трудоемкость различных методов. Естественный, на первый взгляд, критерий – затраченное машинное время – в действительности не подходит: сложно сравнивать быстродействие различных машин, не всегда имеются данные о затратах времени при работе в мультипроцессорном режиме и т.д. Более надежным показателем является число вычислений минимизируемой функции или другая внутренняя характеристика метода. Но и здесь возникает ряд проблем. Например, неясно, как соизмерять трудоемкость вычисления функции и решения различных вспомогательных задач. Затруднительно сопоставлять вычисления функции и ее производных. В тех задачах, где применяются конечно-разностные аппроксимации производных, все намного проще: одно вычисление градиента эквивалентно n вычислениям функции, вычисление гессиана эквивалентно $n(n+1)/2$ вычислениям функции. Однако для дискретной задачи оптимального управления градиент лишь примерно вдвое «дороже» функции [25]. Для квадратичной функции градиент вычисляется даже проще, чем ее значения. Так же дело обстоит и с субградиентом в минимаксной задаче и т.д.

В-третьих, методы могут по-разному вести себя на разных этапах процесса минимизации. Более того, поведение разных величин, характеризующих точность решения, может быть также различным.

Никакого удовлетворительного способа преодоления перечисленных трудностей не существует. Единственное, что можно сделать в подобной ситуации, – приводить данные о результатах вычислений в развернутой форме, чтобы иметь возможность сравнения методов по разным критериям. Рекомендуется при публикации результатов

проверки методов придерживаться следующих правил [25]:

- приводить точную формулировку задачи, для которой проводился счет, включая все ее параметры и начальное приближение;
- указывать тип ЭВМ и данных, язык и среду программирования, сведения о программе;
- давать подробное описание применяемого алгоритма или отсылать к его публикации, если она имеется;
- сообщать различные характеристики точности приближения, невязку в ограничениях и выполнении условий экстремума; в задачах малой размерности приводить и сами приближения;
- указывать подробные сведения о трудоемкости вычислений.

При выполнении этих условий можно воспроизводить полученные данные и сравнивать их с другими по разным показателям.

Далее x^* – точка глобального минимума $F(x)$ на R^n , $F^* = F(x^*)$; k – число итераций;

$\delta F = |F(x^*) - F(x^k)|$ – точность решения по функции; $\delta x = \max |x_i^* - x_i^k|$ – точность решения по аргументам; k_0 – число вычислений функции; k_1 – число вычислений функции (регулировка шага); $\nabla F(x)$ – градиент $F(x)$; $\nabla^2 F(x)$ – матрица вторых производных (гессиан).

Тестовые задачи с известным решением можно построить следующим образом. Выбираются точка x^* и функции $\varphi_i(x)$, $i = 1, 2, \dots, m$. Тогда

$$F(x) = \sum_{i=1}^m (\varphi_i(x) - \varphi_i(x^*))^2 \quad (9)$$

достигает минимума в точке x^* . При этом $F^* = 0$,

$\nabla^2 F(x^*) = \sum_{i=1}^m \nabla \varphi_i(x^*) \nabla^T \varphi_i(x^*)$, так что можно регулировать обусловленность задачи за счет выбора $\varphi_i(x)$. В частности, если $m < n$ или $m = n$, но $\nabla \varphi_i(x^*) = 0$ для некоторых i , получаем вырожденную точку минимума. Функции вида (9), вообще говоря, невыпуклы и могут иметь локальные и даже глобальные минимумы, отличные от x^* .

Другой способ связан с выбором функций вида

$$F(x) = \sum_{i=1}^{n+1} f_i(x_i, x_{i-1}), \quad x_0 = a, \quad x_{n+1} = b. \quad (10)$$

Тогда, задавшись $x_0^* = a$ и решив последовательно одномерные уравнения

$$\partial F / \partial x_i = \partial / \partial x_i (f_i(x_i^*, x_{i-1}^*) + f_{i+1}(x_{i+1}, x_i^*)) = 0 \quad (11)$$

(решение обозначим x_{i+1}^*), для $i = 0, 1, \dots, n$ выберем $b = x_{n+1}^*$. Очевидно, что тогда $\nabla F(x^*) = 0$. Если $f_i(x_i, x_{i-1})$ выпуклы по x_i , x_{i-1} , то $F(x)$ выпукла и x^* – точка минимума $F(x)$.

Наконец, можно взять произвольную гладкую выпуклую функцию $\varphi(x)$ и произвольную точку x^*

и построить $F(x) = \varphi(x) - (\nabla \varphi(x^*), x)$. Тогда $\nabla F(x^*) = 0$, $F(x)$ выпукла и потому x^* – точка глобального минимума $F(x)$. Существуют и другие приемы построения функций с известной точкой минимума.

Следует отметить, что для функций вида (9), (10) существуют специальные методы, обладающие повышенной эффективностью. Например, для задач (11) эффективен метод Гаусса–Ньютона [3], для (11) – метод динамического программирования и покоординатного спуска [25]. Аналогичным образом специфика задач (9), (11) может сказаться на поведении общих методов оптимизации.

Развитие подхода и численные исследования

В [3] доказано, что значение нормы поправки E^k можно дополнительно уменьшить, если использовать симметричные перестановки строк и столбцов H^k на шаге 2 вычислительной схемы 1 модифицированного разложения Холецкого следующим образом. Найти такой индекс q , что $|c_{qq}^k| = \max_{i \leq j \leq n} |c_{ij}^k|$, и поменять местами все данные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Подход Гилла и Мюррея к дополнительному уменьшению нормы поправки E^k развит в [6]. Доказано, что фактическое значение нормы поправки E^k можно дополнительно уменьшить, если использовать симметричные перестановки строк и столбцов H^k . На очередном, i -м, шаге факторизации в качестве i -й строки и i -го столбца надо брать ту из нетронутых пар, для которой величина $|c_{ii}^k| + |c_i^k|$ максимальна.

Такая стратегия дает возможность увеличить численную устойчивость расчета элементов D^k , U^k , u^k и остается работоспособной и в том случае, когда $\|h^k\|$ не равна нулю, но очень мала. Там же отмечено: «Параметр β подбирается путем минимизации априорной оценки нормы поправки E^k и с очевидностью зависит от способа задания направления спуска и, как следствие, от величин c_{ij}^k и g_i^k . Рассмотрение предлагаемой формулы вычисления величины β , учитывающей такую зависимость, не является целью данной работы и находится в стадии экспериментального сравнения эффективности». В данной работе величину β предлагается вычислять по формуле

$$\beta^2 = \max \{ \gamma^0, \rho, \xi / \nu, \varepsilon_M \}, \quad (12)$$

где $\rho = \max_{1 \leq i \leq n} |h_i^k|$, ε_M – машинная точность; ε_M вводится в формулу расчета β , чтобы обеспечить устойчивость вычислений, когда норма H^k очень мала. Применение формулы (12) хорошо согласо-

уется с численными экспериментами, описанными в данной работе. Ниже приводится предлагаемая модификация вычислительной схемы 2, дана наиболее эффективная схема организации расчетов.

Вычислительная схема 3 модифицированного разложения Холецкого. Все фигурирующие в ней величины при реализации на ЭВМ могут размещаться в памяти, первоначально выделяемой для записи матриц H^k и h^k .

Шаг 0. Вычислить $\beta^2 = \max \{ \gamma^0, \rho, \xi / \nu, \varepsilon_M \}$,

где $\rho = \max_{1 \leq i \leq n} |h_i^k|$, $\nu = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ^0

и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k , ε_M –

машинная точность. Положить $\varepsilon_0 = 2^{-\frac{\tau_F}{2}}$,

$\varepsilon_s = 2^{-\frac{\tau_F}{3}} \left(1 + \|F(x^{k-1}) - F(x^k)\| \right)$. Здесь τ_F – число правильных разрядов $F(x^k)$, которые хотелось бы получить.

Шаг 1. Положить $i = 1$ и $c_{jj}^k = h_{jj}^k$ ($j = 1, \dots, n$), $c_j^k = -h_j^k$ ($j = 1, \dots, n$).

Шаг 2. Найти индекс q , такой, что $|c_{qq}^k| = \max_{i \leq j \leq n} (|c_{ij}^k| + |c_j^k|)$, и поменять местами все дан-

ные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Шаг 3. Если $c_{ii}^k < \varepsilon_0$, то положить $l_i^k = \delta$; иначе положить $l_i^k = c_{ii}^k$.

Шаг 4. Если $|c_{ii}^k| \leq \varepsilon_0$, то положить $n_0 = n_0 + 1$ и перейти к шагу 8; иначе перейти к шагу 5.

Шаг 5. Если $c_{ii}^k > 0$, то перейти к шагу 8; иначе перейти к шагу 6.

Шаг 6. Положить $n_0 = n_0 + 1$. Если $\|h^k\| > \varepsilon_s$, перейти к шагу 8; иначе перейти к шагу 7.

Шаг 7. Построить систему уравнений $U^k p^k = e_i$ для поиска направления отрицательной кривизны по формулам $u_j^k = 0$, $j = 1, \dots, n$, $u_i^k = 1$, положить $n_s = n_s + 1$ и остановиться.

Шаг 8. Если $i = n$, то положить $\theta_i = 0$; иначе вычислить вспомогательные величины c_{ij}^k , $j = 1, \dots, n$,

по формуле $c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{is}^k u_{sj}^k d_{ss}^k$ и найти

$$\theta_i = \max_{i+1 \leq j \leq n} |c_{ij}^k|.$$

Шаг 9. Вычислить диагональный элемент фактора D^k и элемент вектора u^k по формулам $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, $u_i^k = c_i^k / d_{ii}^k$. Вычислить $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$). Если $i = n$, перейти к шагу 11.

Шаг 10. Вычислить строку $u_{ij}^k = c_{ij}^k / d_{ij}^k$, $j = 1, \dots, n$, фактора U^k , пересчитать вспомогательные величины по формулам $c_{jj}^k = c_{jj}^k - u_{ij}^k c_{ij}^k$, $j = i+1, \dots, n$, $c_j^k = c_j^k - u_{ij}^k c_i^k$, $j = i+1, \dots, n$, положить $i = i+1$ и перейти к шагу 2.

Шаг 11. Найти $\gamma^k = \max_{1 \leq i \leq n} g_i^k$, положить $uk = \gamma^k u^k$ и остановиться.

В данной работе отмечена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц в [26] и предлагаемого подхода к увеличению эффективности численных методов ньютоновского и квазиньютоновского типов – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских и квазиньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных – это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой. Данное исследование является прямым продолжением работы [14].

Программные реализации. Приложения Nmbm и NmbmApp – машинная реализация ньютоновских методов оптимизации с регулировкой шага, основанных на факторизации Холецкого (вычислительная схема 2 модифицированного разложения Холецкого) [27, 28].

NmbmApp отличается от своего классического прототипа Nmbm конечно-разностной аппроксимацией первых и вторых производных. Все версии алгоритмов реализованы на языке Visual Basic .NET, среда разработки – Microsoft Visual Studio 2010. Исходные тексты программ доступны на сайте <http://crm.ics.org.ru> и могут быть использованы по универсальной общедоступной лицензии GNU.

Приложение NmbmsApp отличается от своего классического прототипа NmbmApp формированием матрицы вторых производных в соответствии с ее структурой. NmbmsApp – машинная реализация ньютоновских методов оптимизации с регулировкой шага, основанных на факторизации Холецкого (вычислительная схема 3 модифицированного разложения Холецкого).

Главное окно программы и пример результатов решения одной из тестовых задач размещены по ссылке http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/6.jpg/.

Достоинством подхода, основанного на применении формул численного дифференцирования, кроме его универсальности, является низкая стоимость подготовки задачи к компьютерному моделированию [29]. От пользователя требуется лишь написание программы для вычисления значения $F(x)$ при заданном x , поэтому реализованные на ос-

нове численных производных методы оптимизации оказываются по существу прямыми методами, то есть методами, не использующими в своей схеме производные от $F(x)$. В то время как результаты численных исследований, приведенные в [14], показывают, что конечно-разностные аналоги проигрывают в точности решения следующих задач оптимизации.

Задача 12 (функция Поляка, среднеквадратичная аппроксимация экспонентами):

$$F(x) = \sum_{j=1}^{10} [\exp(-0.2j) + 2 \exp(-0.4j) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4)]^2,$$

$$F(x) = \sum_{j=1}^{10} [\exp(-0.2j) + 2 \exp(-0.4j) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4)]^2,$$

$$x^0 = (0.5 \ 0 \ 2.5 \ 3)^T, x^* = (1 \ 1 \ 2 \ 2)^T.$$

Функция невыпуклая, с искривленным оврагом, обусловленность велика.

Задача 13 (функция Вуда):

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1),$$

$$x^0 = (-3 \ -1 \ -3 \ -1)^T,$$

$$x^* = (1 \ 1 \ 1 \ 1)^T, F(x^*) = 0.$$

Функция имеет несколько локальных минимумов, это обстоятельство может вызвать преждевременное окончание процесса.

Задача 14 (функция Степенная):

$$F(x) = (10(x_1 - x_2)^2 + (x_1 - 1)^2)^4,$$

$$x^0 = (-1.2 \ 0)^T, x^* = (1 \ 1)^T, F(x^*) = 0.$$

В таблице 1 приводятся результаты численных исследований, выполненных в трех версиях предложенного ньютоновского метода безусловной минимизации: Nmbm, NmbmApp [6, 9, 14, 27, 28] и предлагаемой NmbmsApp.

Таблица 1

Результаты численных исследований задач 12, 13, 14

Table 1

The results of numerical studies for problems 12, 13, 14

Метод	Функция	k	δF	δx	k_0
NmbmApp	Поляка	40	$4.23 \cdot 10^{-33}$	0	1341
NmbmsApp	Поляка	26	$4.23 \cdot 10^{-33}$	0	837
Nmbm	Вуда	13	0	0	36
NmbmApp	Вуда	14	$4.4 \cdot 10^{-27}$	$2 \cdot 10^{-14}$	470
NmbmsApp	Вуда	16	0	0	325
Nmbm	Степенная	12	0	0	163
NmbmApp	Степенная	13	$4.3 \cdot 10^{-61}$	$8 \cdot 10^{-9}$	323
NmbmsApp	Степенная	13	$4.81 \cdot 10^{-61}$	$8 \cdot 10^{-9}$	232

Результаты данного сравнительного исследования позволяют оценить предлагаемую версию ньютоновского метода безусловной оптимизации с регулировкой шага, основанной на факторизации Холецкого с конечно-разностной аппроксимацией

первых и вторых производных (вычислительная схема 3 модифицированного разложения Холесского) как более предпочтительную по сравнению с NmbmApp (вычислительная схема 2 модифицированного разложения Холесского).

Долан и Морэ впервые предложили применение бенчмаркинга в качестве нового инструмента использования статистических данных для сравнения итерационных алгоритмов путем демонстрации профилей производительности [30]. Термин «бенчмаркинг» англоязычный и не имеет дословного перевода на русский язык, происходит от английских слов «bench» (уровень, высота) и «mark» (отметка). Это словосочетание трактуется по-разному: «опорная отметка», «отметка высоты», «эталонное сравнение» и т.д. Как правило, термином «бенчмаркинг» обозначается один из инструментов совершенствования деятельности.

Профилем производительности P_s для метода решения оптимизационной задачи называется функция распределения какого-либо измеримого показателя производительности [30]. Вычисление профилей производительности позволяет визуализировать различия по эффективности нескольких оптимизационных методов, причем для их построения используется отдельная программа [30].

Функция $P_s(\tau)$ определяется следующим образом: $P_s(\tau) = \frac{1}{n_p} \text{size} \{ p \in P : r_{p,s} \leq \tau \}$,

$$r_{p,s} = \frac{t_{p,s}}{\min \{ t_{p,s} : s \in S \}}.$$

Здесь S – множество сравниваемых методов, P – множество решаемых с помощью этих методов задач. Количество элементов в P обозначено через n_p , соответственно n_s – количество элементов в S . В качестве измеримых показателей производительности $t_{p,s}$, как правило, используются число итераций и число вычислений значений функции (или число вычислений значений функции и оценок градиента) [31–33].

В этой технике можно выбрать индекс производительности в качестве меры сравнения среди рассмотренных алгоритмов и проиллюстрировать полученные результаты с помощью профилей производительности.

Бенчмаркинг, как и любой другой инструмент для сравнения итерационных алгоритмов, имеет свои достоинства и недостатки. Чувствительность только к измеримому показателю производительности $t_{p,s}$ и отсутствие удовлетворительного способа сравнения между собой нескольких методов оптимизации по результатам численных экспериментов [25] часто являются причинами разногласий при интерпретации результатов [30]. Кроме того, вычисление профилей производительности требует отбрасывания задач, для которых машинные реализации соответствующих методов потер-

пели неудачу [30], что усложняет процесс сравнения итерационных алгоритмов.

В данной работе, как и в [31–33], все тестовые задачи описаны в [20], все сравниваемые алгоритмы безусловной оптимизации сходятся к одному и тому же локальному решению, сравнению подлежат подходы к заданию направления спуска, результаты решения заносятся в таблицы.

Следует отметить: если объем статистических данных небольшой, то для сравнения итерационных алгоритмов, как правило, достаточно таблиц [32, 33]; иначе для сравнения итерационных алгоритмов используется бенчмаркинг путем демонстрации профилей производительности [30, 31].

В [31–33] завершение итераций производится с использованием одного из следующих критериев останова:

$$\|h^k\| \leq 10^{-5}, \|h^k\| \leq 10^{-6}, \|h^k\| \leq 10^{-6} \|h^0\| \quad (13)$$

соответственно. Отсюда следует отсутствие статистических данных о точности решения тестовых задач, что является недостатком сравнения итерационных алгоритмов. Поэтому в данной работе предлагается приводить и статистические данные о решении с максимальной точностью, а не только данные, соответствующие критериям (13). Результаты численных исследований приведены в таблицах 2 и 3 соответственно.

Таблица 2

Результаты численных исследований,
соответствующие критериям (13)

Table 2

The results of numerical studies that meet the criteria (13)

Функция	n	k	k_1
Extended Rosenbrock function	2	24	84
Extended Rosenbrock function	4	24	82
Extended Rosenbrock function	6	24	72
Wood function	4	16	28
Generalized Rosenbrock function	2	24	84
Generalized Rosenbrock function	3	23	56
Generalized Rosenbrock function	4	25	57
Extended White & Holst function	2	24	26
Extended White & Holst function	4	25	64
Extended White & Holst function	6	26	106
Extended Penalty function	2	10	101
Extended Penalty function	3	14	90
Extended Penalty function	4	13	83
Perturbed Quadratic function	2	5	24
Perturbed Quadratic function	3	5	28
Perturbed Quadratic function	4	5	0
Raydan 1 function	2	5	0
Raydan 1 function	3	5	0
Raydan 1 function	4	5	0
Raydan 2 function	2	5	0
Raydan 2 function	3	5	0
Raydan 2 function	4	5	0
Diagonal 1 function	2	5	22
Diagonal 1 function	3	6	24
Diagonal 1 function	4	6	24
Diagonal 2 function	2	6	22
Diagonal 2 function	3	6	22
Diagonal 2 function	4	7	22
Diagonal 3 function	2	7	22
Diagonal 3 function	3	7	22
Diagonal 3 function	4	7	22

Таблица 3

**Результаты численных исследований
с максимальной точностью**

Table 3

The results of numerical studies with maximum precision

Функция	n	k	k_1
Extended Rosenbrock function	2	18	18
Extended Rosenbrock function	4	19	16
Extended Rosenbrock function	6	18	26
Wood function	4	12	8
Generalized Rosenbrock function	2	18	18
Generalized Rosenbrock function	3	20	18
Generalized Rosenbrock function	4	20	15
Extended White & Holst function	2	21	26
Extended White & Holst function	4	22	20
Extended White & Holst function	6	20	33
Extended Penalty function	2	7	40
Extended Penalty function	3	10	23
Extended Penalty function	4	11	61
Perturbed Quadratic function	2	2	0
Perturbed Quadratic function	3	2	0
Perturbed Quadratic function	4	4	0
Raydan 1 function	2	3	0
Raydan 1 function	3	3	0
Raydan 1 function	4	4	0
Raydan 2 function	2	4	0
Raydan 2 function	3	4	0
Raydan 2 function	4	4	0
Diagonal 1 function	2	3	0
Diagonal 1 function	3	3	1
Diagonal 1 function	4	4	2
Diagonal 2 function	2	4	0
Diagonal 2 function	3	4	0
Diagonal 2 function	4	4	0
Diagonal 3 function	2	4	0
Diagonal 3 function	3	4	0
Diagonal 3 function	4	4	0

Заключение

В работе доказано, что интеграция техники исключения Гаусса и факторизации Холецкого определяет решение проблемы масштабирования шагов при спуске, а следовательно, и аппроксимацию неквадратичными функциями, и интеграцию с методом доверительной окрестности [1], а также то, что интеграция техники исключения Гаусса и факторизации Холецкого определяет и подходы к дальнейшему уменьшению значения нормы поправки E^k , если использовать симметричные перестановки строк и столбцов H^k следующим образом: на очередном, i -м, шаге факторизации в качестве i -й строки и i -го столбца надо брать ту из нетронутых пар, для которой величина $|c_{ii}^k| + |c_i^k|$ максимальна [6].

Кроме того, доказано, что модифицированная факторизация Холецкого – оптимизированный алгоритм в том смысле, что параметр β подбирается в нем путем минимизации априорной оценки нормы поправки E^k при условии сохранения существенно положительно определенной матрицы неизменной и с очевидностью зависит от способа задания направления спуска [1, 6]. В данной работе доказанные утверждения позволили модифициро-

вать формулу вычисления величины β , предложенную в [2], и построить вычислительную схему 3. Полученные результаты хорошо согласуются с численными экспериментами, приведенными в данной работе.

Изучен подход к увеличению эффективности ньютоновских методов с конечно-разностной аппроксимацией первых и вторых производных. Подход является основой для дальнейших исследований, результаты которых могут быть использованы для построения численных методов ньютоновского типа.

Рассмотрена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц [26] и предлагаемого подхода к увеличению эффективности численных методов ньютоновского типа – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регуляризацией шага и с конечно-разностной аппроксимацией первых и вторых производных, это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой. Данное исследование является прямым продолжением работы [14].

Литература

1. Зеленков Г.А., Хакимова А.Б. Подход к разработке алгоритмов ньютоновских методов оптимизации, программная реализация и сравнение эффективности // Компьютерные исследования и моделирование. 2013. Т. 5. № 3. С. 367–377.
2. Gill P.E. and Murray W. Newton-type methods for unconstrained and linearly constrained optimization. Math. Prog. 1974, no. 28, pp. 311–350.
3. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация; [пер. с англ.]. М.: Мир, 1985. 509 с.
4. Гилл Ф., Мюррей У. Численные методы условной оптимизации; [пер. с англ.]. М.: Мир, 1977. 290 с.
5. Парлетт Б. Симметричная проблема собственных значений. Численные методы; [пер. с англ.]. М.: Мир, 1983. 382 с.
6. Свириденко А.Б. Априорная поправка в ньютоновских методах оптимизации // Компьютерные исследования и моделирование. 2015. Т. 7. № 4. С. 835–863.
7. Хакимова А.Б., Зеленков Г.А. Увеличение эффективности ньютоновских методов минимизации. Вычисление длины шага // Динамика неоднородных систем: тр. ИСА РАН. 2010. Вып. 14-А. Т. 53. С. 115–120.
8. Хакимова А.Б., Дикусар В.В., Зеленков Г.А. Увеличение эффективности ньютоновских методов оптимизации. Информационно-динамический подход // Динамика неоднородных систем: тр. ИСА РАН. 2010. Вып. 14-А. Т. 53. С. 97–114.
9. Свириденко А.Б., Зеленков Г.А. Управление процессом построения ньютоновских методов безусловной оптимизации: монография. Новороссийск: Изд-во ГМУ им. адм. Ф.Ф. Ушакова, 2016. 98 с.
10. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. М.: Наука, 1975. 320 с.
11. Полак Э. Численные методы оптимизации. Единый подход. М.: Мир, 1974. 376 с.
12. Карманов В.Г. Математическое программирование. М.: Наука, 1975. 272 с.
13. Stepleman R.S., Winarsky N.D. Adaptive numerical differentiation. Mathematics of Computation, 1979, no. 33, pp. 1257–1264.
14. Свириденко А.Б., Зеленков Г.А. Взаимосвязь и реализа-

ция квазиньютоновских и ньютоновских методов безусловной оптимизации // Компьютерные исследования и моделирование. 2016. Т. 8. №. 1. С. 55–78.

15. Averick B.M., Carter R.G., Moré J.J. The MINPACK-2 test problem collection (Preliminary version). Math. and Comp. Sc. Divis., 1991, no. 150.

16. Averick B.M., Carter R.G., Moré J.J., Xue G.L. The MINPACK-2 test problem collection. Math. and Comp. Sc. Divis., 1992.

17. Jackson R.H.F., and Mulvey J.M. A Critical review of comparisons of mathematical programming algorithms and software (1953–1977). J. Research of the National Bureau of Standards, 1978, no. 83, pp. 563–584.

18. Gillsinn J., Hoffman K., Jackson R.H.F., Leyendecker E., Saunders P., and Shier D. Methodology and analysis for comparing discrete linear L1 approximation codes. Communications in Statistics, 1977, no. 136, pp. 399–413.

19. Crowder H.P., Dembo R.S., and Mulvey J.M. On reporting computational experiments with mathematical software. ACM Transactions on Mathematical Software, 1980, no. 5, pp. 193–203.

20. Andrei N. An unconstrained optimization test functions collection. Adv. Model. Optim. 2008, vol. 1, no. 10, pp. 147–161.

21. Bongartz I., Conn A.R., Gould N.I.M., Toint P.L., CUTE: constrained and unconstrained testing environments. ACM TOMS, 1995, vol. 21, pp. 123–160.

22. Moré J.J., Garbow B.S., Hillstom K.E. Testing unconstrained optimization software. ACM Trans. Math. Soft., 1981, vol. 7, pp. 17–41.

23. Himmelblau D., Applied nonlinear programming. McGraw-Hill, NY, 1972.

24. Lee D. A fast and robust unconstrained optimization method requiring minimum storage. Mathematical Programming, 1985, vol. 32, pp. 41–68.

25. Поляк Б.Т. Введение в оптимизацию. М.: Наука, 1983. 384 с.

26. Соловьев С.А. Решение разреженных систем линейных уравнений методом Гаусса с использованием техники аппроксимации матрицами малого ранга // Вычислительные методы и программирование. 2014. Т. 15. С. 441–460.

27. Свириденко А.Б. MNB (Ньютоновский метод безусловной оптимизации). Свид. о гос. регистр. прогр. для ЭВМ № 2015610399. 2015.

28. Свириденко А.Б., Зеленков Г.А. MNBApp (Ньютоновский метод безусловной оптимизации с численным вычислением первых и вторых производных). Свид. о гос. регистр. прогр. для ЭВМ № 2015610347. 2015.

29. Черноуцкий И.Г. Методы оптимизации. Компьютерные технологии. СПб: БХВ-Петербург, 2011. 384 с.

30. Dolan E., Moré J.J. Benchmarking optimization software with performance profiles. Math. Program., 2002, no. 91, pp. 201–213.

31. Ahookhosh M., Amini K. An efficient nonmonotone trust-region method for unconstrained optimization. Numer Algor, 2012, no. 59, pp. 523–540.

32. Runak M. Abdula, Abbas Y. Al Bayati. A new hybrid scaled search direction for unconstrained optimization. Jour. of Kirkuk University, 2007, vol. 2, no. 1, pp. 62–81.

33. Abbo K.K. New CG Method for large-scale unconstrained optimization based on Nazareth theorem. Iraqi Jour. of Statistical Sciences, 2008, vol. 13, pp. 53–65.

Software & Systems

DOI: 10.15827/0236-235X.118.270-281

Received 10.10.16

2017, vol. 30, no. 2, pp. 270–281

REALIZATION AND TESTING OF NEWTON METHODS FOR UNCONSTRAINED OPTIMIZATION

A.B. Sviridenko¹, Lecturer, roshechka@gmail.com

¹ Novorossiysk branch of the Kuban State University, Geroev Desantnikov St. 87, Novorossiysk, 353900, Russian Federation

Abstract. The paper describes an approach to increasing the effectiveness of Newton's methods for unconstrained optimization based on the Cholesky factorization with step adjustment and finite-difference approximation of the first and second derivatives. The modified Cholesky decomposition of the second derivative matrix is the basis of increasing the efficiency of the Newton's methods. It defines the solution to the problem of scaling steps during lowering; the approximation by non-quadratic functions; integration with the method of confidence neighborhoods; decrease in norms of priori amendment. The work investigates the possibility of reducing the number of function evaluations by forming the second derivative matrix in accordance with its structure.

The article examines the relationship of the approach to increasing the efficiency of the Gaussian exceptions for sparse matrices and the proposed approach to increasing the efficiency of numerical methods of Newton type (using a matrix structure), that is information about matrix positions which store non-zero elements. For Newton's methods for unconstrained optimization based on the Cholesky factorization with step adjustment and the finite-difference approximation of the first and second derivatives it is the possibility of reducing the number of function evaluations by forming a matrix of the second derivatives in accordance with its structure.

The paper contains the descriptions of program implementations, all versions of the algorithms are implemented in Visual Basic .NET, the development environment is Microsoft Visual Studio 2010. The article shows the results of numerical studies of the effectiveness of the developed algorithms given the set of rules described in the work.

The work studies an approach to increasing the efficiency of Newton's methods with finite-difference approximation of the first and second derivatives. This is the basis for further research, the results of which might be used to build Newton-type numerical methods.

Keywords: newton's methods, matrix structure, finite-difference approximation, Cholesky's decomposition, scaling steps, method of confidence neighborhoods, norm of the prior correction, benchmarking, performance profile, collection of test problems.

References

1. Zelenkov G.A., Khakimova A.B. Approach to development of algorithms of Newtonian methods of unconstrained optimization, their software implementation and benchmarking. *Kompyuternye issledovaniya i modelirovaniye* [Computer Research and Modeling]. 2013, vol. 5, no. 3, pp. 367–377 (in Russ.).

2. Gill P.E., Murray W. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Prog.* 1974, no. 28, pp. 311–350.
3. Gill P., Murray U., Wright M. *Practical Optimization*. Academic Press, 1981 (Russ. ed.: Moscow, Mir Publ., 1985, 509 p.).
4. Gill P.E., Murray W. *Numerical methods of constrained optimization*. Academic Press, 1974 (Russ. ed.: Moscow, Mir Publ., 1977, 290 p.).
5. Parlett B. *The Symmetric Eigenvalue Problem*. Numerical methods. Univ. of California, 1980 (Russ. ed.: Moscow, Mir Publ., 1983, 384 p.).
6. Sviridenko A.B. The correction to Newton's methods of optimization. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2015, vol. 7, no. 4, pp. 835–863 (in Russ.).
7. Khakimova A.B., Zelenkov G.A. Increasing the efficiency of Newton's minimization techniques. Calculating a step length. *Trudy ISA RAN "Dinamika neodnorodnykh sistem"* [Proc. ISA RAS "Dynamics of Inhomogeneous Systems"]. Moscow, Librokom Publ., 2010, vol. 53, pp. 115–120 (in Russ.).
8. Khakimova A.B., Dikumar V.V., Zelenkov G.A. Increasing the efficiency of Newton minimization techniques. Informodynamic approach. *Trudy ISA RAN "Dinamika neodnorodnykh sistem"* [Proc. of ISA RAS "Dynamics of Inhomogeneous Systems"]. Moscow, Librokom Publ., 2010, vol. 53, pp. 97–114 (in Russ.).
9. Sviridenko A.B., Zelenkov G.A. *Upravlenie protsessom postroyeniya nyutonovskikh metodov bezuslovnoy optimizatsii* [Managing the Process of Constructing Newton's Methods for Unconstrained Optimization]. Monograph. Novorossiysk, Admiral Ushakov Maritime State Univ. Publ., 2016, 98 p.
10. Pshenichny B.N., Danilin Yu.M. *Chislennyye metody v ekstremalnykh zadachakh* [Numerical methods in Extreme Problems]. Moscow, Nauka Publ., 1975.
11. Polak E. *Computational Methods in Optimization: A Unified Approach*. Academic Press, 1971, 329 p. (Russ. ed.: Ershko F.I. Moscow, Mir Publ., 1974, 376 p.).
12. Karmanov V.G. *Matematicheskoe programmirovaniye* [Mathematical Programming]. Moscow, Mir Publ., 1975.
13. Stepleman R.S., Winarsky N.D. Adaptive numerical differentiation. *Mathematics of Computation*. 1979, no. 33, pp. 1257–1264.
14. Sviridenko A.B., Zelenkov G.A. Correlation and realization of quasi-Newton methods of absolute optimization. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2016, vol. 8, no. 1, pp. 55–78 (in Russ.).
15. Averick B.M., Carter R.G., Moré J.J. *The MINPACK-2 Test Problem Collection* (Preliminary version). Math. and Comp. Sc. Divis. 1991, no. 150.
16. Averick B.M., Carter R.G., Moré J.J., Xue G.L. The MINPACK-2 test problem collection. Math. and Comp. Sc. Divis., 1992.
17. Jackson R.H.F., Mulvey J.M. A Critical Review of Comparisons of Mathematical Programming Algorithms and Software (1953–1977). *J. Research of the National Bureau of Standards*. 1978, no. 83, pp. 563–584.
18. Gillsinn J., Hoffman K., Jackson R.H.F., Leyendecker E., Saunders P., Shier D. Methodology and Analysis for Comparing Discrete Linear L1 Approximation Codes. *Communications in Statistics*. 1977, no. 136, pp. 399–413.
19. Crowder H.P., Dembo R.S., Mulvey J.M. On Reporting Computational Experiments with Mathematical Software. *ACM Trans. on Mathematical Software*. 1980, no. 5, pp. 193–203.
20. Andrei N. An unconstrained optimization test functions collection. *Adv. Model. Optim.* 2008, no. 10 (1), pp. 147–161.
21. Bongartz I., Conn A.R., Gould N.I.M., Toint P.L. *CUTE: Constrained and Unconstrained Testing Environments*. ACM TOMS Publ., 1995, vol. 21, pp. 123–160.
22. Moré J.J., Garbow B.S., Hillstom K.E. Testing unconstrained optimization software. *ACM Trans. Math. Soft.* 1981, vol. 7, pp. 17–41.
23. Himmelblau D. *Applied Nonlinear Programming*. McGraw-Hill, NY, 1972.
24. Lee D. A fast and robust unconstrained optimization method requiring minimum storage. *Mathematical Programming*. 1985, vol. 32, pp. 41–68.
25. Polyak B.T. *Vvedenie v optimizatsiyu* [Introduction to Optimization]. Moscow, Nauka Publ., 1983.
26. Solov'ev S.A. Solution of sparse systems of linear equations using the Gauss method and low-rank approximations matrices. *Vychislitelnye metody i programmirovaniye* [Numerical Methods and Programming]. Moscow, 2014, vol. 15, pp. 441–460 (in Russ.).
27. Sviridenko A.B. *MNB (Nyutonovskiy metod bezuslovnoy optimizatsii)* [MNB (Newton's Method for Unconstrained Optimization)]. State registration Certificate of the computer program № 2015610399. 2015 (in Russ.).
28. Sviridenko A.B., Zelenkov G.A. *MNBApp (Nyutonovskiy metod bezuslovnoy optimizatsii s chislennym vychisleniem pervykh i vtorykh proizvodnykh)* [MNBApp (Newton's Method for Unconstrained Optimization with Numerical Calculation of the First and Second Derivatives)]. State registration Certificate of the computer program № 2015610347. 2015 (in Russ.).
29. Chernorutsky I.G. *Metody optimizatsii. Kompyuternyye tekhnologii* [Optimization Methods. Computer Technologies]. St. Petersburg, BHV-Peterburg Publ., 2011, 384 p.
30. Dolan E., Moré J.J. Benchmarking optimization software with performance profiles. *Math. Program.* 2002, no. 91, pp. 201–213.
31. Ahookhosh M., Amini K. An efficient nonmonotone trust-region method for unconstrained optimization. *Numer. Algor.* 2012, no. 59, pp. 523–540.
32. Abdula R.M., Al Bayati A.Y. A new hybrid scaled search direction for unconstrained optimization. *Jour. of Kirkuk Univ.* 2007, vol. 2, no. 1, pp. 62–81.
33. Abbo K.K. New CG Method for Large-Scale Unconstrained Optimization Based on Nazareth theorem. *Iraqi Jour. of Statistical Sciences*. 2008, vol. 13, pp. 53–65.

УДК 519.722

DOI: 10.15827/0236-235X.118.282-285

Дата подачи статьи: 08.11.16

2017. Т. 30. № 2. С. 282–285

ЭФФЕКТИВНЫЙ МЕТОД БЛОЧНОГО КОДИРОВАНИЯ ДВУХУРОВНЕВЫХ ИЗОБРАЖЕНИЙ

М.П. Бакулина, к.ф.-м.н., научный сотрудник, marina@rav.sccc.ru

(Институт вычислительной математики и математической геофизики СО РАН,
просп. Академика Лаврентьева, 6, г. Новосибирск, 630090, Россия)

Эффективное сжатие изображений без потери качества – одна из важных задач теории информации, имеющая широкое практическое применение. Известно, что любое цифровое изображение можно представить последовательностью сообщений. Для выбора таких сообщений существуют многочисленные способы. Единственное требование к ним – обеспечение возможности восстановления точной копии исходного изображения из последовательности сообщений.

Один из способов выбора сообщений состоит в том, что смежные элементы изображения группируются в блоки, которые затем кодируются в соответствии с вероятностями их появления. При этом для наиболее вероятных конфигураций блоков используются короткие кодовые слова, а для менее вероятных – длинные, в результате чего в среднем достигается сжатие данных. Такой тип кодирования называется блочным.

Применяя метод блочного кодирования, можно добиться эффективного сжатия без потери качества двухуровневых (бинарных) изображений, где каждый элемент изображения может быть либо черным, либо белым. Решению этой задачи и посвящена данная статья.

Любое двухуровневое изображение рассматривается как множество примыкающих друг к другу прямоугольных блоков определенного размера. Эти блоки образуют совокупность сообщений, характеризующую изображение. Применяя к совокупности блоков оптимальный код Хаффмана, можно добиться наибольшего сжатия данных. Однако для блоков размером более 3×3 совокупность сообщений оказывается очень большой, и использование кода Хаффмана становится неэффективным. Кроме того, статистический анализ конфигураций блоков для двухуровневых изображений показывает, что блок, состоящий из белых элементов, имеет достаточно высокую вероятность.

Исходя из этого наблюдения и применяя известный оптимальный код, в данной работе предлагается эффективный двухэтапный метод блочного кодирования для двухуровневых изображений. Найдены оптимальные размеры блока, используемые на первом этапе кодирования, а также проведено сравнение экспериментальных результатов сжатия построенного алгоритма с результатами сжатия известного блочного алгоритма JPEG, которое подтвердило эффективность предложенного метода.

Ключевые слова: двухуровневое изображение, блочное кодирование, степень сжатия.

Эффективное сжатие изображений без потери качества – одна из важных задач теории информации, имеющая широкое практическое применение. Ее решению посвящен ряд исследований (например [1–3]). Любое цифровое изображение можно представить последовательностью сообщений. Для выбора таких сообщений существуют многочисленные способы. Единственное требование к ним – обеспечение возможности восстановления точной копии исходного изображения из последовательности сообщений. Один из способов выбора сообщений заключается в том, что смежные элементы изображения группируются в блоки размером $n \times m$, где n и m – число элементов в горизонтальном и вертикальном направлениях соответственно. Затем полученные блоки кодируются в соответствии с вероятностями их появления, причем для наиболее вероятных конфигураций блоков используются короткие кодовые слова, а для менее вероятных – длинные кодовые слова, в результате чего в среднем достигается сжатие данных. Такой тип кодирования называется блочным и рассматривается в работах [4, 5]. Блочное кодирование с использованием адаптивных методов рассмотрено в работах [6, 7].

Любое двухуровневое (бинарное) изображение мы будем рассматривать как совокупность примы-

кающих друг к другу прямоугольных блоков размером $n \times m$. Каждый элемент изображения может быть либо черным (1), либо белым (0), поэтому общее число конфигураций блоков, то есть число возможных расположений элементов в пределах блока, составляет 2^{nm} . Эти блоки образуют совокупность сообщений, которая характеризует изображение. Применяя к совокупности блоков оптимальный код Хаффмана [8], можно добиться наибольшего сжатия данных. Однако для блоков, размер которых более 3×3 , совокупность сообщений оказывается очень большой, и использование кода Хаффмана становится нецелесообразным.

В данной работе предлагается эффективный двухэтапный метод блочного кодирования для двухуровневых изображений. Найдены оптимальные размеры блока на первом этапе кодирования, а также проведено сравнение экспериментальных результатов сжатия построенного алгоритма с результатами сжатия блочного алгоритма JPEG [9], подтверждающее эффективность предложенного метода.

Метод блочного кодирования

Статистический анализ конфигураций блоков для двухуровневых изображений показывает, что

блок, состоящий из белых элементов, имеет достаточно высокую вероятность. Исходя из этого наблюдения и предложенного в [10] субоптимального кода, построим эффективный метод блочного кодирования. Кодирование будем осуществлять в два этапа.

Рассмотрим первый этап. Кодовое слово для блока, состоящего из одних нулей, будет 0. Кодовые слова для других конфигураций блоков образуются из nm двоичных разрядов, соответствующих данному блоку, которым предшествует префикс 1. Распределение вероятностей для источника, при котором этот код оптимален, равно

$$P(i, n, m) = \begin{cases} P(0, n, m) & \text{при } i = 1, \\ \frac{1 - P(0, n, m)}{2^{nm} - 1} & \text{при } i = 2, \dots, 2^{nm}, \end{cases} \quad (1)$$

где $P(0, n, m)$ – вероятность полностью белого блока.

Тогда средняя длина кодового слова определяется в виде

$$L = P(0, n, m) + (1 + nm)(1 - P(0, n, m)) = nm(1 - P(0, n, m)) + 1. \quad (2)$$

Пусть теперь $y_1 y_2 \dots y_l$ – последовательность, полученная после первого этапа кодирования, $y_i \in \{0, 1\}$. Рассмотрим второй этап кодирования, осуществляемый арифметическим кодом из [10]. Обозначим через $p = p(1)$ и $q = p(0)$. Выделим в этой

последовательности серии длины $l = \left\lfloor \frac{1}{\sqrt{p}} \right\rfloor$, кото-

рые следуют после появления 1, и особые символы $\underline{0}$ и $\underline{1}$, не входящие в блоки, то есть представим последовательность $y_1 y_2 \dots y_l$ в виде

$$\underline{0} \dots \underline{0} \underline{1} \underbrace{y_1 \dots y_l}_{i-1} \underline{0} \dots \underline{0} \underline{1} \underbrace{y_1 \dots y_l}_i.$$

Кодирование различных y_i осуществляется арифметическим кодом из [10] с помощью различных кодеров, настроенных на различные вероятности появления нулей и единиц, и может быть описано следующим образом. Особые символы $\underline{0}$ и $\underline{1}$ кодируются с помощью кодера K_0 с вероятностями q^l и $1 - q^l$ для $\underline{0}$ и $\underline{1}$ соответственно. Рассмотрим кодирование символов, находящихся внутри блока $y_1 \dots y_l$ длины l . Пусть $y_1 \dots y_{i-1} = \underbrace{0 \dots 0}_{i-1}$ ($i = 1, \dots, l$).

Тогда символ y_i , находящийся в i -й позиции после появления $i - 1$ нулей, кодируется с помощью кодера K_i с вероятностями π_i и $(1 - \pi_i)$ для $\underline{0}$ и $\underline{1}$ соответственно, где

$$\pi_i = \frac{1 - q}{1 - q^{l-i+1}} = \frac{1}{1 + q + q^2 + \dots + q^{l-i}}. \quad (3)$$

Появление этих вероятностей объяснено в [11]. Наконец, символы в блоке $y_1 \dots y_l$, следующие после появления в этом блоке 1, кодируются с помощью кодера \bar{K} с исходными вероятностями q и p для 0 и 1 соответственно. Важно отметить, что ве-

роятности π_i не хранятся в памяти кодера и декодера, а вычисляются по следующей рекуррентной формуле:

$$\frac{1}{\pi_{i+1}} = \frac{1}{\pi_i} - q^{l-i}. \quad (4)$$

Следовательно, вычисление π_i , определяемых формулой (3), можно организовать по следующей схеме: $\bar{q} := \bar{q} / q$, $\bar{\pi}^{-1} := \bar{\pi}^{-1} - \bar{q}$ с начальными данными $\bar{q} := q^l$, $\bar{\pi}^{-1} := 1 + q + \dots + q^{l-1}$.

Таким же образом кодируется следующий блок, причем перед каждым новым блоком начальные данные обновляются. Для нахождения оптимальных размеров блока на первом этапе кодирования найдем теоретический коэффициент сжатия, полученный после первого этапа. Под коэффициентом сжатия C будем понимать отношение числа двоичных разрядов, необходимых для представления заданного изображения до кодирования, к числу двоичных разрядов после кодирования. Учитывая (2), получаем

$$C = \frac{nm}{\bar{L}} = \frac{nm}{nm(1 - P(0, n, m)) + 1}. \quad (5)$$

Ограничимся рассмотрением квадратных блоков $n \times n$. В таблице 1 приведены результаты зависимости коэффициента сжатия, полученные теоретически $C_{\text{теор}}$ и экспериментально $C_{\text{эксп}}$, от размера квадратного блока n для различных двухуровневых изображений A1–A5. Отметим, что для теоретических и экспериментальных результатов размер n был взят в интервале от 2 до 6. Это объясняется тем, что для $n > 6$ коэффициент сжатия начинает уменьшаться, поэтому дальнейшее увеличение размера блока становится нецелесообразным.

Таблица 1

Зависимость теоретических и экспериментальных результатов коэффициента сжатия от размера квадратного блока

Table 1

Dependence of theoretical and experimental results of a compression ratio on the size of a square block

Изображение	Коэффициент сжатия	Размер блока, n				
		2	3	4	5	6
A1	$C_{\text{теор}}$	3,01	5,22	6,91	7,15	6,87
	$C_{\text{эксп}}$	3,26	5,54	7,13	7,44	7,08
A2	$C_{\text{теор}}$	3,25	5,19	6,26	6,61	6,30
	$C_{\text{эксп}}$	3,15	5,06	6,11	6,54	6,19
A3	$C_{\text{теор}}$	3,21	5,15	5,68	5,60	5,24
	$C_{\text{эксп}}$	3,06	4,89	5,53	5,48	5,07
A4	$C_{\text{теор}}$	2,99	3,68	3,91	3,81	3,57
	$C_{\text{эксп}}$	2,83	3,56	3,81	3,66	3,48
A5	$C_{\text{теор}}$	2,52	3,23	3,31	3,14	2,95
	$C_{\text{эксп}}$	2,45	3,14	3,19	3,03	2,87

Из таблицы 1 видно, что наилучшие значения коэффициента сжатия дают размеры $n = 4$ и $n = 5$, которые и являются оптимальным размером блока.

Сравнение экспериментальных результатов сжатия

Для подтверждения эффективности предложенного метода было проведено сравнение экспериментальных результатов сжатия построенным алгоритмом с результатами наиболее распространенного и известного стандарта сжатия изображений – блочным методом JPEG. В качестве тестовых изображений были взяты рассмотренные выше двухуровневые изображения A1–A5. Сравнение проводилось по степени сжатия. Под степенью сжатия в данном случае понимаем количество бит, которым представляется в сжатом файле один байт (8 бит) исходного (несжатого) изображения. Результаты степени сжатия для предложенного метода k_{NEW} и для блочного метода JPEG – k_{JPEG} представлены в таблице 2.

Таблица 2

Результаты степеней сжатия различных двухуровневых изображений для предложенного алгоритма и известного блочного метода JPEG

Table 2

The results of a compression ratio of various binary images for the proposed algorithm and the famous JPEG decomposition method

Изображение	k_{NEW}	k_{JPEG}
A1	2,94	3,16
A2	4,58	4,81
A3	3,53	3,77
A4	4,91	5,14
A5	5,15	5,39

Из таблицы видно, что степень сжатия k_{NEW} примерно на 22–24 % лучше степени сжатия блочным методом JPEG, что подтверждает эффективность предложенного метода.

Выводы

Полученные экспериментальные данные показывают эффективность предложенного в данной работе метода блочного кодирования двухуровневых изображений: он дает сжатие таких изображений на 22–24 % лучше, чем известный и широко применяемый метод JPEG. Показано, что оптимальные размеры блока для предложенного метода – 4×4 или 5×5 . Построенный алгоритм кодирования может быть использован на практике для эффективного сжатия картографических и факсимильных изображений, спутниковых изображений земной поверхности и т.д.

Литература

1. Todd S., Langdon G.G., Rissanen J. Parameter reduction and context selection for compression of the gray-scale images. IBM Jour. Res. Develop, 1985, vol. 29, no. 2, pp. 188–193.
2. Howard P.G., Vitter J.S. Fast and efficient lossless image compression. Proc. IEEE Data Compression Conf. Snowbird, Utah, USA, 1993, pp. 351–360.
3. Babu P., Sathappan S. Efficient lossless image compression using modified hierarchical prediction and context adaptive coding. Indian Jour. of Sc. and Tech., 2015, vol. 8, pp. 1–6.
4. Tan D.S., Turner L.F. Sequential block interleave coding of two-tone facsimile data. Computers and Digital Techniques, 1988, vol. 38, pp. 95–107.
5. Horlander F.J. Incremental scanning for facsimile. IBM Tech. Disc. Bull, 1972, vol. 14, pp. 3311–3313.
6. Coulon F., Jonsen O. Adaptive block scheme for source coding of black-and-white facsimile. Electron. Lett., 1976, vol. 12, no. 3, pp. 61–62.
7. Bracamonte J., Ansorge M., Pellandini F. Adaptive block-size transform coding for image compression. IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing, 1997, vol. 4, pp. 2721–2724.
8. Кормен Т.Х., Лейзерсон Ч.И., Ривест Л.Р., Штайн К. Алгоритмы: построение и анализ. М.: Вильямс, 2007. 459 с.
9. Weinberger M.J., Seroussi G., Sapiro G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Trans. Image Process, 2000, vol. 9, no. 8, pp. 1310–1324.
10. Ryabko B.Ya., Fionov A.N. Homophonic coding with logarithmic memory size. Algorithms and Computation, Springer, Berlin, 1997, pp. 253–262.
11. Рябко Б.Я., Шарова М.П. Быстрое кодирование низкоэнтропийных источников // Проблемы передачи информации. 1999. Т. 35. № 1. С. 49–60.

AN EFFICIENT BLOCK CODING METHOD FOR BINARY IMAGES

M.P. Bakulina¹, Ph.D (Physics and Mathematics), Research Associate, marina@rav.sssc.ru

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Academician Lavrentev Ave. 6, Novosibirsk, 630090, Russian Federation

Abstract. Efficient image compression without quality loss is one of important problems of the information theory. This problem has a wide practical application. It is known that any digital image can be represented by a sequence of messages. There are numerous ways to select messages. The only requirement for these methods is the possibility of restoring an exact copy of the original image from a sequence of messages.

One of the ways of choosing messages is that the adjacent picture elements are grouped into blocks. Then, these blocks are encoded according to the probabilities of their occurrence. Thus, short code words are used for the most probable configurations of blocks, and long code words are used for less probable configurations. The result is an average ratio of data compression. This coding is called block coding.

The block coding method makes it possible to obtain efficient compression two-level (binary) images without losing quality. This paper considers the solution of this problem.

We consider any binary image as many adjacent rectangular blocks of a certain size. These blocks form a set of messages, which characterizes the image. Using the optimal Huffman coding we can achieve the highest data compression. However, in blocks larger than 3×3 a set of messages is very large, and the Huffman code becomes inefficient. In addition, statistical analysis of two-level images shows that a block consisting of white elements has high probability.

Based on this observation and applying the known optimal code, this paper proposes an efficient two-step block coding method for binary images. We found the optimal block size at the first stage of coding. We have also compared the experimental results of the compression ratio with the proposed algorithm and the block algorithm JPEG. The results have confirmed the efficiency of the proposed method.

Keywords: binary image, block coding, compression ratio.

References

1. Todd S., Langdon G.G., Rissanen J. Parameter reduction and context selection for compression of the gray-scale images. *IBM Jour. Res. Develop.* 1985, vol. 29, no. 2, pp. 188–193.
2. Howard P.G., Vitter J.S. Fast and efficient lossless image compression. *Proc. IEEE Data Compression Conf.* Snowbird, Utah, 1993, pp. 351–360.
3. Babu P., Sathappan S. Efficient lossless image compression using modified hierarchical prediction and context adaptive coding. *Indian Jour. of Science and Technology.* 2015, vol. 8 (34), pp. 1–6.
4. Tan D.S., Turner L.F. Sequential block interleave coding of two-tone facsimile data. *Computers and Digital Tech-nique.* 1988, vol. 38, pp. 95–107.
5. Horlander F.J. Incremental scanning for facsimile. *IBM Tech. Disc. Bull.* 1972, vol. 14, pp. 3311–3313.
6. Coulon F., Jonsen O. Adaptive block scheme for source coding of black-and-white facsimile. *Electron. Lett.* 1976, vol. 12, no. 3, pp. 61–62.
7. Bracamonte J., Ansoorge M., Pellandini F. Adaptive block-size transform coding for image compression. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing.* 1997, vol. 4, pp. 2721–2724.
8. Cormen T.H., Leiserson Ch.E., Rivest R.L., Stein C. *Introduction to Algorithms.* 2nd ed., MIT Press & McGraw-Hill Publ., 2001 (Russ. ed.: Moscow, Vilyams Publ., 2007, 459 p.).
9. Weinberger M.J., Seroussi G., Sapiro G. The LOCO-I lossless image compression algorithm: Principles and standarti-zation into JPEG-LS. *IEEE Trans. Image Process.* 2000, vol. 9, no. 8, pp. 1310–1324.
10. Ryabko B.Ya., Fionov A.N. Homophonic coding with logarithmic memory size. *Algorithms and Computation.* Berlin, Springer Publ., 1997, pp. 253–262.
11. Ryabko B.Ya., Sharova (Bakulina) M.P. Fast coding of low-entropy sources. *Problemy peredachi informatsii* [Problems of Information Transmission]. 1999, vol. 35, no. 1, pp. 49–60 (in Russ.).

Примеры библиографического описания статьи

1. Бакулина М.П. Эффективный метод блочного кодирования двухуровневых изображений // Программные продукты и системы. 2017. Т. 30. № 2. С. 282–285; DOI: 10.15827/0236-235X.118.282-285.
2. Bakulina M.P. An efficient block codig method for binary images. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 282–285 (in Russ.); DOI: 10.15827/0236-235X.118.282-285.

УДК 519.677

DOI: 10.15827/0236-235X.118.286-290

Дата подачи статьи: 07.02.17

2017. Т. 30. № 2. С. 286–290

ТЕКСТУРИРОВАНИЕ ТРЕХМЕРНОЙ МОДЕЛИ, ПОСТРОЕННОЙ ПО ОБЪЕДИНЕННОМУ НАБОРУ ВИДОВ С ЗАШИВКОЙ ДЫР

А.П. Кудряшов, к.т.н., м.н.с., kudryashovA@dvo.ru
(Институт автоматизации и процессов управления ДВО РАН,
ул. Радио, 5, г. Владивосток, 690041, Россия);

И.В. Соловьев, аспирант, igorek.soloviyev@mail.ru
(Институт прикладной математики ДВО РАН, ул. Радио, 7, г. Владивосток, 690041, Россия)

В работе предложен автоматический метод генерации и нанесения текстуры на компьютерную трехмерную модель, построенную с помощью воксельного метода объединения триангуляционных сеток, полученных по дальностным данным с разных видов. Этот метод объединения основан на воксельном или объемном подходе, который также использует диффузное размытие значений неявной весовой функции в воксельном пространстве для зашивки дыр на финальной модели. Обычная проекция текстур образует заметные швы на границах сканов, а все рассмотренные аналоги текущего метода обладают высокой вычислительной сложностью. Представленный метод текстурирования предлагает решение этих проблем. Он основан на слиянии всех доступных текстур с учетом веса каждого отдельного пикселя в каждом треугольнике, перекрытия и наблюдаемости этих треугольников. Вес каждого пикселя учитывает следующие параметры: угол, под которым он наблюдается, тень/пересвет, дистанция от края скана, поскольку именно на краю скана часто образуется видимый шов текстур. Для уменьшения времени работы предложены алгоритм индексации треугольников в виде текстурной карты, который позволяет избежать полного перебора треугольников, алгоритм для быстрого расчета весов треугольника относительно дистанции этого треугольника от края скана, а также многоядерные вычисления. Вычислительные эксперименты были проведены на реальных сценах, реконструированных с помощью трехмерного сканера. Они показали, что текущий алгоритм способен генерировать реалистичные текстуры за короткое время для высокодетализированных моделей.

Ключевые слова: текстурирование, воксельное пространство, трехмерная реконструкция, триангуляционная оболочка, объемный метод.

Большая часть методов построения трехмерных моделей реальных объектов использует дальностные данные, получаемые с помощью стереокамер, лазерных дальномеров или алгоритмов компьютерного зрения по серии изображений этого объекта, снятого с разных точек. В научной литературе на такие данные ссылаются как на дальностные изображения (range images) или карты глубин (depth maps) [1]. С каждым пикселем range image/depth map связывается расстояние до видимой точки объекта в пространстве сцены. Пространственные данные могут быть представлены в виде облака точек в трехмерном пространстве или в виде триангуляционной поверхности. Наряду с задачей объединения этих триангуляционных сеток, подразумевающей построение единой связной оболочки без дублирования участков поверхности, стоит задача зашивки дыр и построения текстуры. Зашивка дыр происходит на этапе заполнения неявной функции в воксельном пространстве, хотя методы зашивки могут использоваться на этапе обработки триангуляционной модели, как, например, в [2, 3].

Представленный в настоящей статье метод построения единой целостной трехмерной модели основывается на подходе, предложенном в [4, 5], а также на более ранних работах [6, 7] и направлен на преодоление указанных недостатков. Работа выполнялась в контексте решения более общей задачи по созданию трехмерных сцен на последовательностях изображений. Вклад авторов состоит в разработке оригинального алгоритма, в котором осуществляется построение гибридной весовой

функции для слияния текстур с учетом таких факторов, как угол наблюдения, тень/пересвет и близость к краю скана, обеспечивается оптимизация вычислений за счет эффективной структурной организации данных, а также достигается высокая скорость обработки данных за счет параллельных вычислений на многоядерных процессорах.

Обзор существующих методов

Существует довольно много подходов к текстурированию реконструированных объектов, однако для каждого метода реконструкции с его ограничениями необходимо подбирать свой метод текстурирования. В предложенном методе подразумевается, что сканер стоит неподвижно, а модель вращается на специальной площадке, поэтому для каждого участка поверхности меняется освещение, а, например, в [8] модель стоит неподвижно, а сканер перемещается вокруг, поэтому модель освещена одинаково для разных видов, но необходимо убрать погрешности сканирования. В [9] авторы пользуются низкополигональными объектами и, помимо прочего, решают задачу частичного перекрытия треугольников, а в рассматриваемом случае это модель с треугольниками, текстура которых имеет площадь, близкую к площади одного-двух пикселей, и решать задачу частичного перекрытия нет смысла. Но в целом эти методы близки к предложенному в данной статье.

В [8] используется блендинг с равновесным участием всех видов. Это хорошо работает, если

освещение модели меняется незначительно и не появляются дополнительные тени. Предложенный в настоящей статье метод дает лучший результат, чем равновесный блендинг. В [9] используется блендинг с учетом веса по наклону нормали от вектора наблюдения и веса по глубине, но не учитывается вес по дистанции от края скана, а именно на краю скана обычно образуются треугольники с максимальными искажениями, которые могут дать некорректные результаты. Также не учитывается вклад теней и пересветов.

Стоит отметить и скорость работы. Для [8] это 10–30 минут для 10–20 тысяч треугольников, для [9] – 1–4 минуты для 10–15 тысяч треугольников. Предложенным в данной работе методом обрабатывается за 0.5–1.5 минуты 2–5 миллионов треугольников, то есть обработка осуществляется на несколько порядков быстрее. Однако из-за метода съемки в нашем случае нет необходимости устранять неточности при сканировании.

Также стоит отметить работу [10], в которой для обеспечения нужной точности моделирования и текстурирования используется, помимо фотокамеры, еще и лазерный сканер.

Построение модели с зашивкой дыр

Подробное описание метода построения объединений модели приведено в [3], зашивки дыр – в [4], текстурирования – в [11]. Исходными данными являются набор триангуляционных сеток, полученных стереометодом с разных точек наблюдения, и фотоизображения с известной внешней и внутренней калибровкой камер. В результате необходимо получить объединенную текстурированную модель (рис. 1). Сканирование выполнено трехмерным сканером фирмы RangeVision.

Объединение происходит с помощью воксельного метода, использующего октодереву, для оптимизации применения компьютерной памяти. Основная метода – построение гибридной весовой

функции в каждом прилегающем к поверхности модели вокселе. Данная функция строится на самом нижнем, детальном уровне октодеревы, а затем пересчитывается на четыре уровня вверх, где происходит диффузное размытие. Диффузное размытие необходимо для заполнения пустых вокселей значениями с соседних (непустых) вокселей, по этим значениям будет строиться новая поверхность. Построение итоговой поверхности производится методом марширующих кубиков.

Наиболее ресурсоемкой частью представленного выше метода является определение расстояний от вокселей до ближайшей поверхности при построении весовой функции. Вычислительная трудоемкость возникает из-за необходимости перебора всех треугольников с вычислением соответствующих расстояний. Чтобы избежать избыточных вычислений, предлагается механизм с предварительной индексацией всех треугольников сетки каждого вида. Для этого используется двухмерный массив с размерностью, равной размерности изображения этого вида.

Тогда можно спроецировать каждый треугольник исходной сетки i -го вида на i -е изображение. В результате каждое значение массива приобретает либо индекс, указывающий на принадлежность к определенному треугольнику, либо 0, то есть отсутствие проекции. Такая процедура возможна, поскольку исходная сетка для каждого вида является заведомо видимой, то есть не содержит невидимых на этом виде граней. Использование индексов при последующей работе алгоритма позволяет избежать полного перебора по всем треугольникам каждого вида и существенно уменьшает время работы алгоритма.

Текстурирование

Предварительная фильтрация. Первая проверка – проверка на наблюдаемость. Каждый треугольник построенной трехмерной модели проецируется на каждый из первоначальных видов. Для каждого из них рассчитывается угол наклона от нормали треугольника к линии, проходящей через центр камеры. Если угол превышает 90° , треугольник невидим с этого вида и его можно отбросить. Для треугольников, у которых этот угол близок к 90° , велика вероятность ошибки, но на случай, если нет другого, более подходящего вида, эта проекция будет учтена и отбросится только в случае наличия

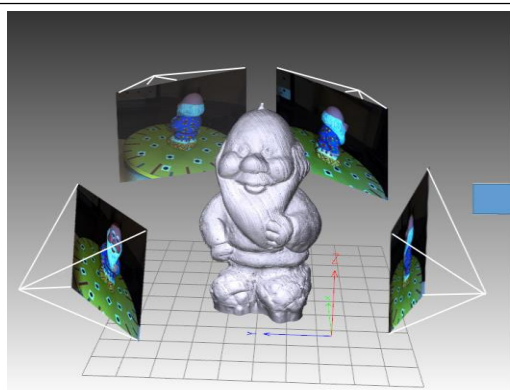


Рис. 1. Построение единой текстурированной трехмерной модели по набору изображений, полученных с разных видов

Fig. 1. Unified textured 3D modeling using images from different views

лучшей альтернативы. Вторая проверка – проверка на видимость, то есть на то, что наблюдаемый треугольник не перекрыт другими треугольниками этой модели. В общем случае потребовался бы перебор всех треугольников модели, что заняло бы значительное время. Для оптимизации можно воспользоваться уже построенной для первоначальной модели индексной картой. Для всех углов треугольника объединенной модели получим значения индексов треугольников исходных сканов до объединения. Если расстояние между этими объектами меньше порогового значения, значит, не существует объектов, которые загромождали бы рассматриваемый треугольник. Особняком стоят треугольники, полученные в результате зашивки дыр: у них нет индексов на первоначальной индексной карте, поэтому либо придется проверять их на видимость путем полного перебора, либо, если необходимо сэкономить время, игнорировать этот критерий, а использовать только критерий наблюдаемости. Логично, что для текстуры каждого треугольника нужно выбрать видимый участок, снятый под минимальным углом. Однако на стыках текстур, полученных с разных видов, появляются заметные границы (рис. 2а). Рассмотрим подробнее решение этой задачи.

Попиксельное смешивание. Основной проблемой при текстурировании трехмерной модели с помощью исходных фотоизображений является различное освещение на разных видах. На рисунке 3 продемонстрировано, как один и тот же участок поверхности модели выглядит по-разному на разных видах.

Решить эту задачу можно путем смешивания значений цвета для каждой точки текстуры с разных видов. Однако равномерное смешивание текстур приводит к не самым идеальным результатам

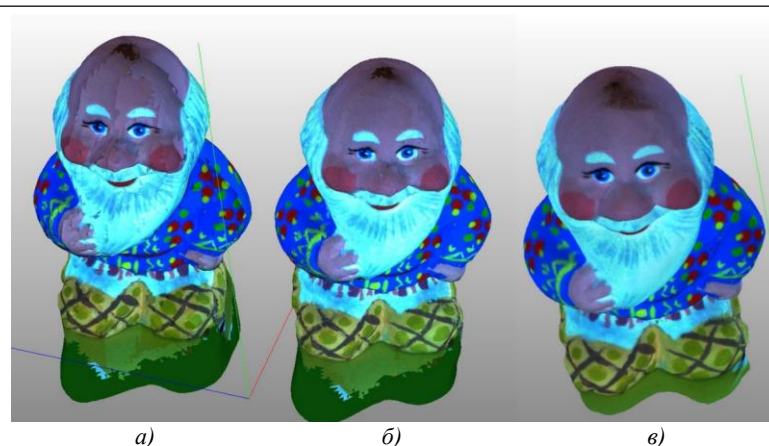


Рис. 2. Разные типы нанесения текстур: а) по лучшим критериям, б) равномерный блендинг, в) блендинг с гибридным весом

Fig. 2. Different types of texturing: а) according to the best criteria, б) uniform blending, в) hybrid weight blending

(рис. 2б). Дело в том, что виды неравнозначны, поскольку на некоторых из них присутствуют сильные тени и пересветы, а на некоторых нужный треугольник снят под очень большим углом или находится на границе скана, где обычно могут быть ошибки сканирования. Поэтому для учета всех предложенных критериев выбора текстуры был использован гибридный вес, рассчитываемый по формуле

$$W = w_{\text{осв}}(w_{\text{уг}} + w_{\text{гр}}), \quad (1)$$

где $w_{\text{осв}}$ – вес по освещенности (функция имеет трапециевидальную форму, принимает значение 0 для теней, затем линейно возрастает до 1 и также линейно падает до 0 для пересветов); $w_{\text{уг}}$ – вес по углу наблюдения (линейно возрастает от 0 до 1 в зависимости от угла наблюдения); $w_{\text{гр}}$ – вес по расстоянию треугольника от границы скана.

Если первые два параметра не вызывают трудностей с расчетом, то расчет расстояния от каждого треугольника до границы скана в прямом виде является крайне ресурсоемким, поэтому была предложена следующая оптимизация. Уже существует построенная структура индексов для всех треугольников, которая в двухмерном виде содержит

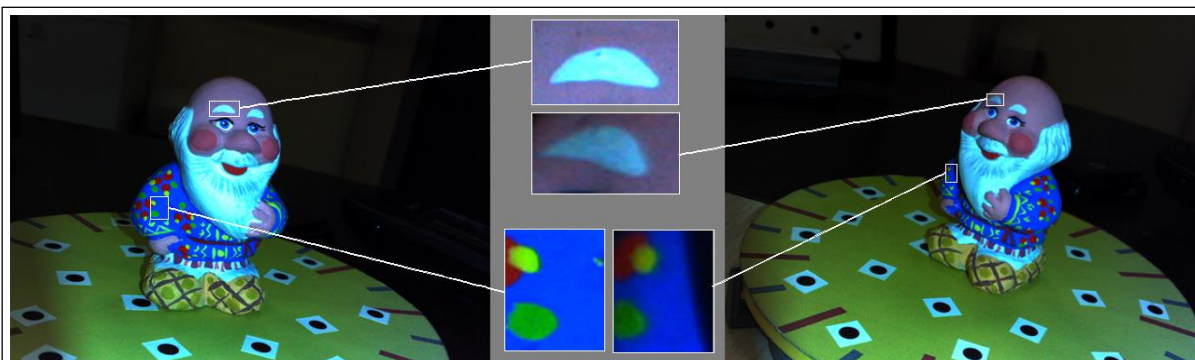


Рис. 3. Влияние освещения на текстуру

Fig. 3. Lighting effect on a texture

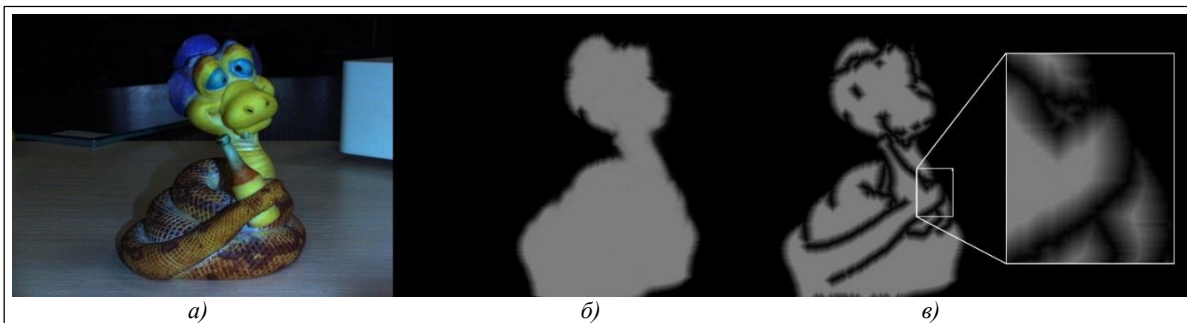


Рис. 4. Фотоизображение модели «Змея» (а), карта весов по дистанции от края скана (б), гибридная карта весов с учетом всех весов (в)

Fig. 4. "Snake" model photo image (a), weight map on a distance from the edge of a scan (b), hybrid weight map adjusted for all weights (c)

индексы треугольников и нули там, где треугольников нет. Необходимо для каждого ненулевого значения вычислить расстояние до ближайшего нуля. Поскольку найденные значения используются только в качестве веса, нет необходимости в вычислении абсолютных значений в 3D, достаточно вычислить дистанцию на плоскости. Однако прямой перебор всех ненулевых значений и поиск ближайшего нулевого в заданном радиусе занимают продолжительное время, около 1.2 сек. на каждый вид. Поэтому была предложена следующая оптимизация. Каждому ненулевому значению индекса присваиваем максимальный вес, затем проверяем, есть ли рядом нулевое значение; если есть, добавляем в список и присваиваем значение 1. Затем осуществляем поиск значений, равных максимальному весу, только в соседних ячейках элементов из этого списка. Такую итерацию производим необходимое число раз. Данная оптимизация позволяет сократить время вычисления веса в 6–8 раз.

На рисунке 4 для одного вида модели «Змея» (а) показаны построенная карта весов по дистанции от края скана (б), а также гибридная карта весов с учетом всех весов: по освещению, углу и расстоянию до края скана (в), рассчитанная по формуле (1).

Рассчитав вес для каждого пикселя, построим финальную текстуру. Для каждого пикселя каждого треугольника вычисляем цвет по формуле

$$C = \sum_{i=0}^n W_i c_i, \quad (2)$$

где n – число видов; c_i – исходный цвет с i -го вида; W_i – вес, который должен быть нормирован так, чтобы сумма всех весов для всех видов была равна 1. Результат текстурирования представлен на рисунке 2в.

Результаты

Для оценки эффективности предложенного метода проведены вычислительные эксперименты на реальных сценах, полученных с помощью сканера RangeVision 3D (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/9.jpg). Использование

многоядерности позволило увеличить скорость работы в среднем в 2,5 раза по сравнению с однопроцессорным вариантом. Результаты экспериментов отражены в таблице.

Сводная таблица результатов

Summary of results

Параметр	Модель	
	Дед	Змея
Количество треугольников в исходной модели, млн	1,8	2,7
Количество видов	8	10
Время работы алгоритма реконструкции, сек.	18	37
Время работы алгоритма текстурирования, сек.	6	12
Количество треугольников в финальной модели, млн	0.3	0.6

Заключение

В работе представлен метод текстурирования триангуляционной трехмерной модели, построенной воксельным методом с зашивкой дыр. Тестирование на реальных данных показало, что предложенные алгоритмические решения и структура данных с использованием октодерев, индексации треугольников, вычисления гибридной весовой функции для каждой точки текстуры существенно повысили качество визуализации и производительность метода по сравнению с аналогами.

Работа выполнена при частичной финансовой поддержке РФФИ (гранты №№ 15-07-00341, 16-07-00350), Программы «Дальний Восток» (проект 15-I-4-011 о) и Программы Президиума РАН № I.33П «Фундаментальные проблемы математического моделирования».

Литература

1. Ramamoorthi R., Arvo J. Creating generative models from range images. Proc. SIGGRAPH'99, 1999, pp. 195–204.
2. He X.J., Chen Y.H. A Haptics-guided hole-filling system based on triangular mesh. Computer Aided Design and Application. 2006, vol. 6, no. 3, pp. 711–718.
3. Jun Y. A piecewise hole filling algorithm in reverse engi-

neering. *Computer-Aided Design*. 2005, vol. 2, no. 37, pp. 263–270.

4. Curless B., Levoy M. A volumetric method for building complex models from range images. *Proc. SIGGRAPH'96*. 1996, pp. 303–312.

5. Davis J., Marschner S.R., Garr M., Levoy M. Filling holes in complex surfaces using volumetric diffusion. *Proc. 1st Intern. Sympos. on 3D Data Processing, Visualization, and Transmission*. Padua, Italy, 2005, pp. 428–438.

6. Бобков В.А., Кудряшов А.П. Воксельный метод построения триангуляционной поверхности по множеству видов // Информатика и системы управления. 2012. № 2. С. 31–38.

7. Кудряшов А.П., Черкашин А.С. Построение единой триангуляционной поверхности по набору видов с зашивкой дыр // Информатика и системы управления. 2015. № 1. С. 36–40.

8. Gal R., Wexler Y., Ofek E., Hoppe H., Cohen-Or D. Seamless montage for texturing models. *Computer Graphics Forum*. 2009, vol. 2, no. 29, pp. 479–486.

9. Baumberg A. Blending images for texturing 3D models. *Proc. British Machine Vision Conference (BMVC)*, 2002, pp. 112–126.

10. Alshawabkeh Y., Haala N. Automatic multi-image photo-texturing of complex 3D scenes. *Proc. XVIII CIPA Int. Sympos.*, Torino, Italy, 2005, pp. 68–73.

11. Кудряшов А.П., Соловьев И.В. Создание единой текстурированной трехмерной модели по набору видов // Программные продукты, системы и алгоритмы. 2017. № 1. URL: <http://swsys-web.ru/creating-a-textured-three-dimensional-model-on-a-set-of-species.html> (дата обращения: 06.02.2017).

Software & Systems

DOI: 10.15827/0236-235X.118.286-290

Received 07.02.17

2017, vol. 30, no. 2, pp. 286–290

TEXTURING OF 3D MODELS WITH FILLED HOLES

A.P. Kudryashov¹, Ph.D. (Engineering), Junior Researcher, kudryashovA@dvo.ru

I.V. Solov'ev², Postgraduate Student, igorek.solovyev@mail.ru

¹ Institute of Automation and Control Processes Far Eastern Branch of RAS, Radio St. 5, Vladivostok, 690041, Russian Federation

² Institute of Applied Mathematics Far Eastern Branch of RAS, Radio St. 7, Vladivostok, 690041, Russian Federation

Abstract. The paper presents an automatic method for texturing three-dimensional computer models. The proposed method is based on the well-known voxel approach, according to which the resulting surface of the scene is obtained by combining a variety of views in the voxel space. Diffuse blurring of weighting function values in the voxel space is used to fill holes. Simple image projection results in visible seams on scan edges. To overcome this problem one normally needs to resort to highly computationally demanding methods. The proposed method of texturing offers a solution for these problems. The texturing algorithm is based on the blending of all available textures, taking into account the weight of each pixel in each triangle, as well the overlap and observability of these triangles. The algorithmic implementation of the method is optimized to improve computational performance of the method and the quality of a generated textured triangulation model. The algorithm of indexing of triangles makes use of a texture map for each view which allows reducing the amount of computation time and therefore to speed up reconstruction and texturing. The multi-core processing provided an additional increase in computational performance. Computational experiments were performed on real three-dimensional scene data obtained by a 3D scanner RangeVision. The experiments showed that the current algorithm is able to generate realistic textures in a short time for highly detailed models.

Keywords: texturing, voxel space, 3D reconstruction, triangulation mesh, volumetric method.

Acknowledgements. The work has been partially financially supported by RFBR (grant no. 15-07-00341, 16-07-00350), a program “Far East” (project 15-I-4-011 o) and the Program of the RAS Presidium no. 1.331I “Fundamental problems of mathematical modeling”.

References

1. Ramamoorthi R., Arvo J. Creating generative models from range images. *Proc. SIGGRAPH'99*. 1999, pp. 195–204.
2. He X.J., Chen, Y.H. A Haptics-guided hole-filling system based on triangular mesh. *Computer Aided Design and Application*. 2006, vol. 6, no. 3, pp. 711–718.
3. Jun Y. A Piecewise hole filling algorithm in reverse engineering. *Computer-Aided Design*. 2005, vol. 2, no. 37, pp. 263–270.
4. Curless B., Levoy M. A Volumetric Method for Building Complex Models from Range Images. *Proc. SIGGRAPH'96*. 1996, pp. 303–312.
5. Davis J., Marschner S.R., Garr M., Levoy M. Filling holes in complex surfaces using volumetric diffusion. *Proc. 1st Int. Symp. on 3D Data Processing, Visualization, and Transmission*. Padua, Italy, 2005, pp. 428–438.
6. Bobkov V.A., Kudryashov A.P. A volumetric method for building triangulated surface from range images. *Informatika i sistemy upravleniya* [Information Science and Control Systems]. 2012, no. 2, pp. 31–38 (in Russ.).
7. Kudryashov A.P., Cherkashin A.S. Building a unified triangulated surface by type sets with filled holes. *Informatika i sistemy upravleniya* [Information Science and Control Systems]. 2015, no. 1 (43), pp. 36–40 (in Russ.).
8. Gal R., Wexler Y., Ofek E., Hoppe H., Cohen-Or D. Seamless montage for texturing models. *Computer Graphics Forum*. 2009, no. 29 (2), pp. 479–486.
9. Baumberg A. Blending images for texturing 3D models. *British Machine Vision Conference – BMVC*. 2002, pp. 112–126.
10. Alshawabkeh Y., Haala N. Automatic multi-image photo-texturing of complex 3D scenes. *Proc. 18th CIPA Int. Symp.* Torino, Italy, 2005, pp. 68–73.
11. Kudryashov A.P., Solov'ev I.V. Creating a unified textured 3D type set model. *Programmnye produkty, sistemy i algoritmy* [Software Products, Systems and Algorithms]. 2017, no. 1. Available at: <http://swsys-web.ru/creating-a-textured-three-dimensional-model-on-a-set-of-species.html> (accessed February 6, 2017).

УДК 519.681.3

DOI: 10.15827/0236-235X.118.291-300

Дата подачи статьи: 10.11.16

2017. Т. 30. № 2. С. 291–300

ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ МОДУЛЬНОЙ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВИАЦИОННОЙ ТЕМАТИКИ

*Л.А. Чижикова, ведущий специалист, l_chizhikova@scac.ru
(Компания «Гражданские самолеты Сухого»,
ул. Поликарпова, 23б, стр. 2, г. Москва, 125284, Россия)*

Разработка ПО – довольно сложный и трудоемкий процесс, в котором проектирование корректной и надежной архитектуры (структуры) играет ключевую роль. Распределение и координация усилий по созданию ПО в группе разработчиков часто оказываются наиболее ответственными и трудными решениями, так как влияют на основной результат.

С развитием технологий для функциональности и управления бортовым радиоэлектронным оборудованием требуется увеличение объемов работ по созданию и сопровождению ПО. В сфере производства авиационных тренажеров и контрольно-проверочной техники роль проектирования, разработки и сопровождения ПО также возрастает. Все большая часть задач переносится с аппаратной части на ПО.

В статье дается анализ основных аспектов проектирования ПО авиационного назначения, сопоставляются принципы проектирования архитектуры ПО для бортового оборудования гражданского самолета и архитектуры ПО авиационного тренажера. Для представления единой модели архитектуры ПО исследованы работы по проектированию архитектуры ПО авиационного тренажера и архитектура бортового ПО самолета. Проведено сравнение подходов, выявлены их общности. Также рассмотрена модель, успешно применяемая при проектировании ПО компанией Rockwell Automation, исследован ряд работ Делфтского технического университета по рассматриваемой тематике.

В работе предлагается математическое представление модульной архитектуры программного продукта, ориентированного на использование в авиационной индустрии.

Предложенный подход к проектированию ПО для применения в авиационной отрасли позволяет унифицировать разработку программных продуктов, сократить временные затраты и трудоемкость их создания, вносить инновационные решения без трансформирования существующей структуры при условии, что программный продукт создавался с применением описанного решения.

Ключевые слова: ПО, архитектура ПО, проектирование архитектуры ПО, разработка ПО, авиационные тренажеры, авиационное ПО, математическое моделирование.

Проектирование ПО является неотъемлемой частью любого проекта, основополагающим фундаментом будущей разработки.

Программная реализация алгоритмов взаимодействия программных модулей и их информационный обмен на аппаратном уровне обеспечивают корректное функционирование разрабатываемой системы.

В группе разработчиков необходимо правильно распределять и координировать усилия по созданию ПО, при этом большое значение имеет четкое определение структуры (архитектуры) разрабатываемой программы [1].

Часто при изначально некорректном проектировании структуры программного продукта на этапе разработки или сопровождения возникает проблема регрессии программного кода, когда дальнейшее развитие и сопровождение невозможны.

Целями исследования способов проектирования архитектуры ПО были выявление, оптимизация и унификация предложенных решений по разработке ПО для применения в авиационной отрасли.

В результате изучения опыта применения модульной структуры ПО компанией Rockwell Automation, исследования зарубежных научных источников, описывающих возможность применения модульной архитектуры при разработке ПО, а

также анализа применяемых на практике технологий разработки ПО в авиационной отрасли было установлено, что основным аспектом качественного проектирования ПО для применения в рассматриваемой области является модульность. Такой подход обеспечивает гибкость и наращиваемость разрабатываемого продукта, легкость в сопровождении, экономии ресурсов.

Методы проектирования архитектуры ПО авиационного тренажера

Рассмотрим методы проектирования архитектуры ПО на примере авиационного полнопилотажного тренажера.

Использование модульной архитектуры ПО в тренажерах предполагает имитацию каждой физической системы в отдельном программном блоке.

При модульной архитектуре построения программные блоки могут создаваться независимо друг от друга и объединяться в систему для получения необходимых результатов [2].

Решением задачи корректного взаимодействия программных модулей является применение алгоритмов передачи данных от модуля к модулю. Данная задача включает как реализацию логических связей между программными блоками, так и обмен информацией системы с взаимодействующими подсистемами, при этом в архитектуре ПО необхо-

димо выделить программный модуль (модули) интерфейсного обмена [2].

При проектировании архитектуры всей системы авиационного тренажера его составляющие (программные блоки) условно можно распределить по уровням взаимодействия. Как правило, в архитектуре ПО выделяют пять уровней распределения программных блоков [1, 3].

В работе [1] представляется архитектура ПО авиационного тренажера в виде программных блоков или модулей, иерархически распределенных по уровням. Там же дано определение архитектуры ПО как структуры пакета программ моделирования самолета.

На самом верхнем уровне находятся программные блоки, предназначенные для трансформации и представления результата всей выработанной (полученной) на более низких уровнях информации в интерактивном виде. На данном уровне содержатся и подпрограммы, описывающие графический пользовательский интерфейс. Это структура, ориентированная на пользователя и содержащая информацию о подпрограммах, которые должны быть выполнены, управление заданиями. Уровень предназначен для представления, подготовки и перекomпоновки всей внутренней информации в интерактивное представление для пользователя, а также для человеко-машинного взаимодействия.

Следующий, второй, уровень включает в себя начальные условия вычислений и фактический процесс моделирования. Кроме того, в начало и в конец соответственно включены две дополнительные части, где данные считываются или хранятся и где удобно осуществлять преобразования данных и перевод информации.

При моделировании полета, а также при использовании методов программной реализации в реальном времени используется цикл с отсчетом времени. Третий уровень содержит вычисления вектора состояния производной по времени и выходного вектора полной системы с приращением. На этом уровне осуществляется обмен информацией: полный результат входных сигналов, вычисленный из разных источников, передается на более низкие программные уровни.

На четвертом, программном, уровне программные модули содержат фактическую математическую модель самолета, в то время как программные модули первых трех уровней не зависят от его типа и модели. Программные модули четвертого уровня имеют доступ к БД и массивам, содержащимся на пятом уровне.

Последний, пятый, уровень содержит БД, массивы информации, необходимые для вычислений на более высоких уровнях, отдельный набор подпрограмм (программных модулей) для интерполяции данных в таблицах, конфигурационной информации, поступления новых данных.

Примером применения описанной архитектуры является реализация аэродинамических моделей тренажера самолетов Boeing 747 и Fokker F28.

Самолет можно представить как относительно малую структуру, движущуюся сквозь атмосферу под действием внешних сил и моментов. Математическая формулировка этих отношений между самолетом и внешними аэродинамическими силами и моментами обычно описана в аэродинамической модели в виде программного модуля.

Так как движение частиц когерентных систем может быть выражено обобщенно, математические модели различных типов *воздушного судна* (ВС), в первую очередь, отличаются соответствующими аэродинамическими моделями.

Первоначально данные для построения аэродинамической модели Fokker F28 были представлены в виде графов. Такое представление требует преобразований в удобную для программирования форму, в этом случае хранят отдельные наборы табличных представлений данных. Промежуточные точки могут быть найдены путем численной интерполяции [1].

Математическая модель была также линеаризована из набора ссылочных состояний с использованием процедуры численной линеаризации. Кроме того, привносятся перекрестные ссылки на оригинальное графическое представление для широкого набора исходных условий, представление линейных систем, допускается дальнейшее упрощение аэродинамической модели путем устранения незначительных отрицательных эффектов. Поскольку аэродинамические модели содержатся в отдельных программных модулях, их легко можно скопировать в целевые приложения, особенно для целевых применений с похожей модульной структурой. Обновления аэродинамической модели могут быть реализованы и путем обновления отдельно хранящихся коэффициентов или замены отдельного модуля.

Примером модульного подхода к построению ПО при разработке авиационных тренажеров [2] может служить структурная схема модуля имитатора динамики полета авиационного тренажера Ту-204, представленная на рисунке 1, где приняты следующие обозначения: ВСС-85 – вычислительная система самолетовождения, ВСУП-85 – вычислительная система управления полетом, ВСУТ-85 – вычислительная система управления тягой, СЭИ-85 – система электронной индикации, ИК – истинный курс, ПВПП – признак касания ВПП.

В данном примере тренажера имитируется движение самолета в пространстве посредством решения замкнутой системы нелинейных дифференциальных уравнений в соответствующих программных блоках, где входными параметрами являются управляющие воздействия экипажа, а выходными – вычисленные параметры полета [4].

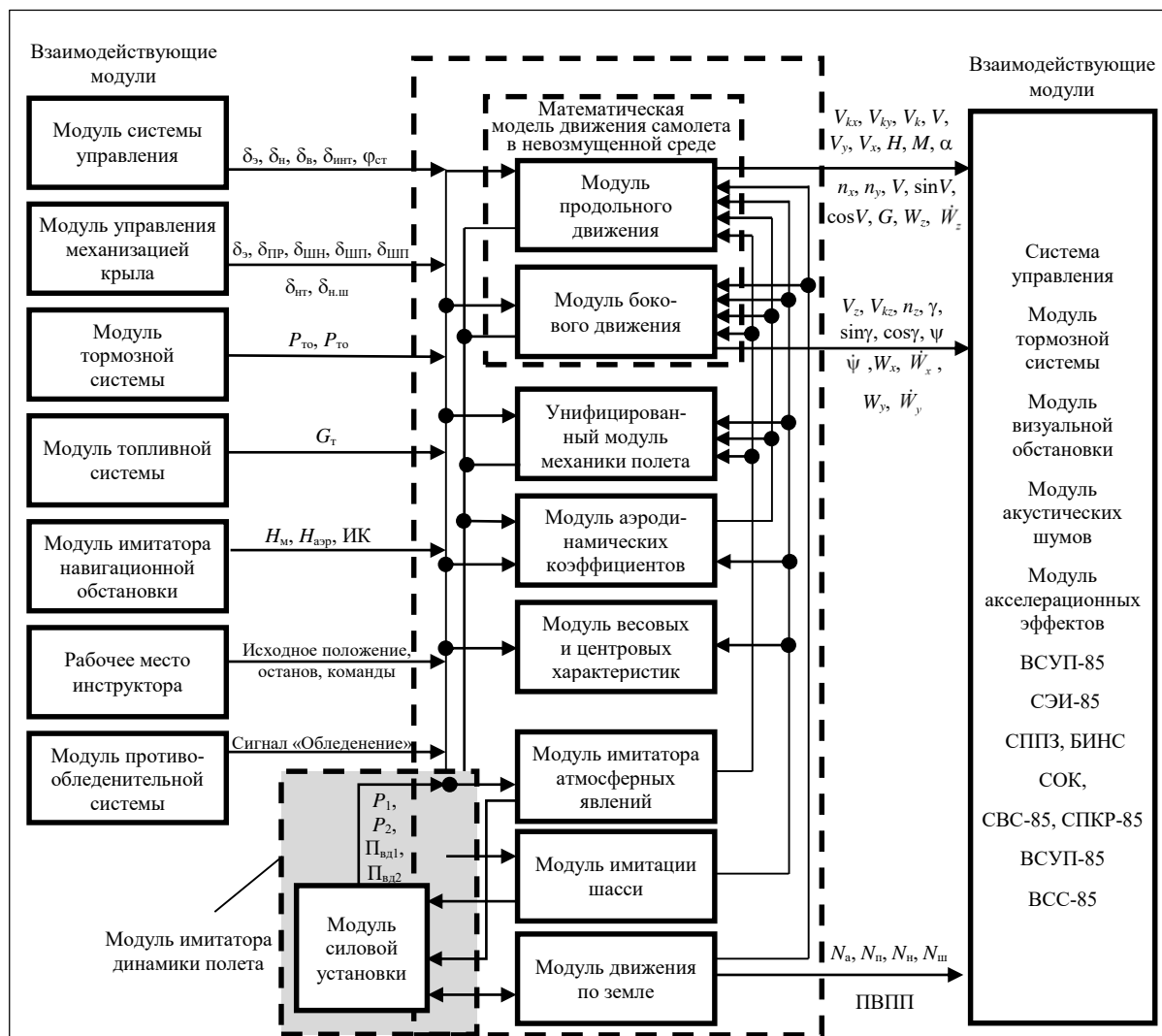


Рис. 1. Структурная схема динамики полета с указанием взаимосвязей с другими моделями

Fig. 1. A flight dynamics structural scheme indicating interactions with other models

Для обеспечения подобия реального и моделируемого полетов самолета осуществляется непрерывное вычисление параметров полета в реальном масштабе времени. Экипажу от соответствующих модулей предоставляется визуальная, акустическая, акселерационная информация, в том числе показания приборов пилотажно-навигационного комплекса, положения и усилия на органах управления (формируемых в соответствии с параметрами, получаемыми в имитаторе динамики полета). Производится декомпозиция системы уравнений движения (продольное и боковое движение, движение по земле, модуль аэродинамических коэффициентов). Шаг интегрирования задается из условий устойчивости и мощности ЭВМ [2].

В работе [5] приведен пример схемы ПО авиационного тренажера, где программные блоки распределены по пяти уровням (рис. 2).

Рассмотрим назначение программных блоков на примере симуляции ВС Airbus серии A300-600.

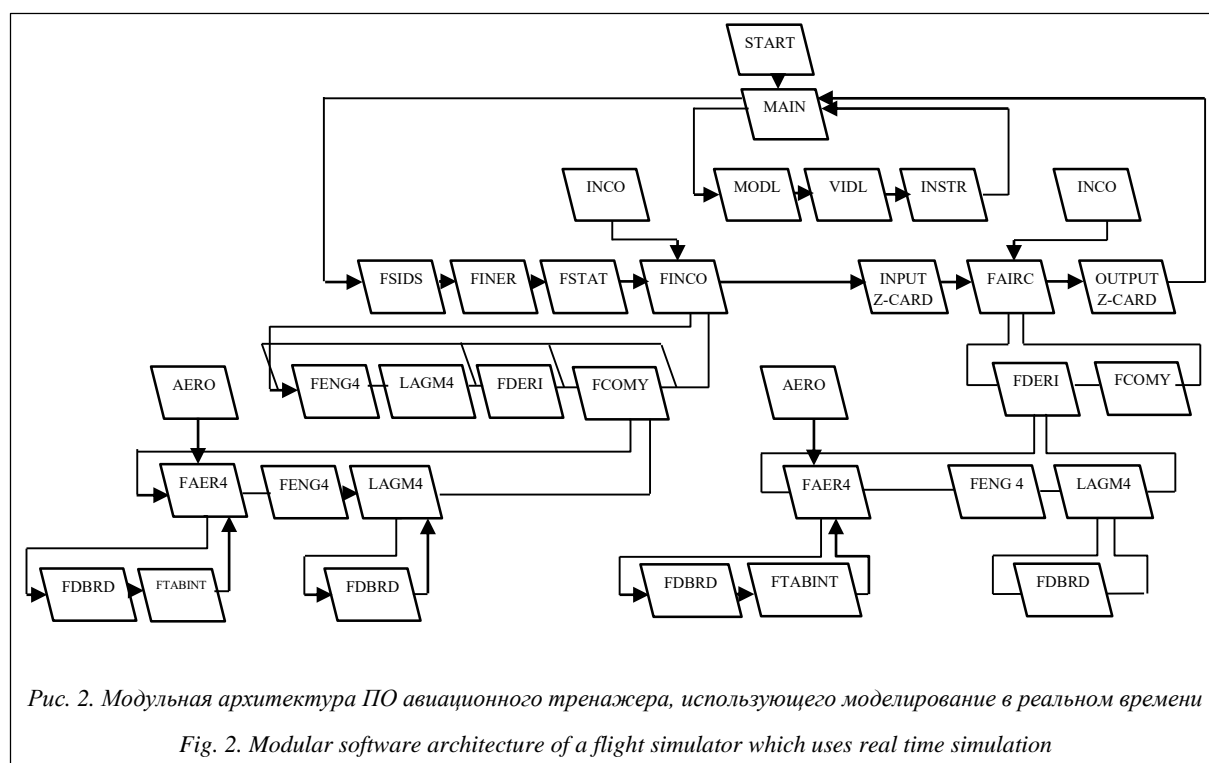
Компьютер GouldSEL 32/87 – сервер, запускающий визуальные сцены для тренажера. Все программные модули в системе тренажера распределены по уровням взаимодействия и представления информации.

Опишем программные блоки первого уровня архитектуры ПО тренажера.

START. Главный программный блок загружаемого программного модуля MAIN.LM. Используется для объявления массива и передачи данных в главную подпрограмму.

MAIN. Управляющая программа (программный модуль) для нелинейной симуляции полета в реальном времени.

INCO. Главный программный блок загружаемого программного модуля INCO.LM. Используется для выделения памяти для обмена информацией между модулями MAIN.LM и INCO.LM, а также между INCO.LM и AERO.LM. В фазе инициализации начальные условия вычисляются посред-



ством модуля FINCO, в процессе симуляции выполняется интеграционный модуль FAIRC.

MODL. Содержит законы управления движением системы подвижности тренажера.

VIDL. Содержит законы управления, генерирующие входы (CGI-Computer Generated Imagery) для визуальной системы авиационного тренажера.

INSTR. Интерфейс инструментов ПО для запуска пилотажных приборов внутри авиационного тренажера.

Назовем программные блоки второго уровня.

FSIDS. Подпрограмма, обеспечивающая набор входных параметров для главного программного модуля MAIN и определяющая описания переменных самолета.

FINER. Подпрограмма, рассчитывающая параметры инерции в уравнениях движения. Для лучшего понимания процессов, происходящих при моделировании, в работе [5] приведены следующие математические уравнения движения самолета в связанной системе координат:

$$\begin{aligned} F_x &= -W \cdot \sin\theta + X = m \cdot (\dot{u} + qw - rv); \\ F_y &= W \cdot \cos\theta \cdot \sin\varphi + Y = m \cdot (\dot{v} + ru - pw); \\ F_z &= W \cdot \cos\theta \cdot \cos\varphi + Z = m \cdot (\dot{w} + pv - qu); \\ M_x &= L = I_x \cdot \dot{p} + (I_z - I_y)qr - J_{xz}(\dot{r} + pq); \\ M_y &= M = I_y \cdot \dot{q} + (I_x - I_z)rp - J_{xz}(p^2 - r^2); \\ M_z &= N = I_z \cdot \dot{r} + (I_y - I_x)pq - J_{xz}(\dot{p} - rq), \end{aligned}$$

где F_x, F_y, F_z – силы, действующие по координатным осям X, Y, Z ; W – вес самолета; θ – угол наклона; X, Y, Z – компоненты аэродинамической силы, приложенной по соответствующим координатным осям; φ – угол крена; $\dot{u} = u/v$ – отношение

компоненты u скорости ЛА, направленной вдоль оси X , к скорости ЛА; q – атмосферное давление; w – компонента скорости ЛА вдоль оси Z ; r – угловая скорость вдоль оси Z ; v – компонента скорости ЛА вдоль оси Z ; M_x, M_y, M_z – моменты сил, направленные вдоль координатных осей X, Y, Z ; I_x, I_y, I_z – моменты инерции, направленные вдоль координатных осей X, Y, Z ; J_{xz} – центробежный момент инерции.

FSTAT. Подпрограмма, рассчитывающая плотность воздуха на заданной высоте полета, использующая модель стандартной атмосферы.

FINCO. Подпрограмма, рассчитывающая начальные условия и использующая нелинейные уравнения движения самолета. Данный программный блок применяется прежде всего для актуальной симуляции.

INPUT Z-CARD. Интерфейс информационного обмена между Gould SEL32/87 и системами симуляции полета. Программный блок Z-CARD конвертирует 32-битные сигналы в корректный выходной сигнал.

FAIRC. Подпрограмма, рассчитывающая вектор состояния ВС (aircraft state vector) и использующая один из трех интеграционных методов: метод Адамса (конечноразностный многшаговый метод численного интегрирования дифференциальных функций), метод Гойна (Heun) и метод Рунге-Кутты.

OUTPUT Z-CARD. Подпрограмма, преобразующая входные сигналы от системы управления полетом в бортовую систему тренажера (после преобразования аналоговых сигналов в цифровые) в 32-разрядные сигналы для Gould 32/87.

Перечислим программные модули третьего, четвертого и пятого уровней.

AERO. Главная процедура данного программного модуля. Aero.LM используется для выделения области памяти для обмена информацией между подпрограммами INCO.LM и AERO.LM.

FENG4. Подпрограмма, рассчитывающая безразмерные силы и моменты, полученные от работы двигателей самолета. Номер показывает тип самолета.

LAGM4. Подпрограмма, рассчитывающая силы и моменты во время режимов авиатакси, взлета и посадки.

FDERI. В ней рассчитывается приращение времени производных самолета с использованием модели аэродинамики (FAIR4), модели двигателя (FENG4), модели шасси (LAGM4).

FCOMY. Подпрограмма, рассчитывающая воздушную скорость V , угол атаки α , угол скольжения β , угол наклона траектории γ , скорость набора (потери) высоты C и высоту H из вектора состояния $x(t)$ и выдающая выходной вектор $y(t)$.

FAIR4. Подпрограмма, вычисляющая аэродинамические силы и моменты, действующие на самолет.

FDBRD. Подпрограмма, считывающая матрицу или вектор из БД, которая содержит, например, коэффициент подъемной силы (C_l) как нелинейную функцию угла атаки (α), положения (позиции) закрылков, числа Маха (M).

FTABINT. Подпрограмма, представляющая линейную интерполяционную таблицу в виде таблиц $M \times N$ или $1 \times N$.

Таким образом, из рассмотренных примеров можно сделать вывод, что при проектировании архитектуры ПО авиационных тренажеров используется модульный метод, где каждый логический блок взаимодействует с другим посредством программного описания интерфейса обмена данными, действиями с входной и выходной информацией и может быть выражен в нескольких программных модулях. Все программные модули расположены по уровням взаимодействия в соответствии с логической структурой.

Описание архитектуры ПО самолета

Рассмотрим методы проектирования архитектуры ПО для ВС.

Традиционно техническое обслуживание ВС воспринимается на основе аппаратных средств. ПО часто рассматривается как часть аппаратного оборудования, которое при обслуживании не требует дополнительного персонала и усилий.

Исторически первоначально использование программных средств для управления и функционирования бортового оборудования самолета не требовалось.

Необходимость в использовании электроники в авиации возникла во время Второй мировой войны. Развитие бортовых *радиолокационных станций* (РЛС) с использованием магнетрона и связанных с ними технологий происходило в стремительном темпе [6].

В конце 1950–начале 1960-х гг. транзисторы вытеснили термоионные клапаны для многих применений. В военной боевой авиации улучшенная экономическая эффективность транзисторов привела к разработке в 1960–1970-х гг. цифровых систем ВС. Они использовались для систем навигации и атаки.

Развитие электронных ламп позволило создать цифровые ЭВМ, но за счет огромного количества аппаратных средств [7].

Во время Второй мировой войны британскими специалистами по криптографии была создана машина под названием Колосс (Colossus), в которой в большом количестве использовались электронные лампы с термокатодом (вакуумные трубки) для выполнения логических операций и вычислений. Машина была огромной и практически неприменимой для полетов.

Первым самолетом, разработанным с использованием цифровых технологий, был A-5 Vigilante, бомбардировщик ВМС США, введенный в эксплуатацию в 1960-е гг.

В конце 1970–начале 1980-х гг. цифровые технологии стали все шире применяться как в авиационных системах управления, так и в системах боевых действий (рис. 3).

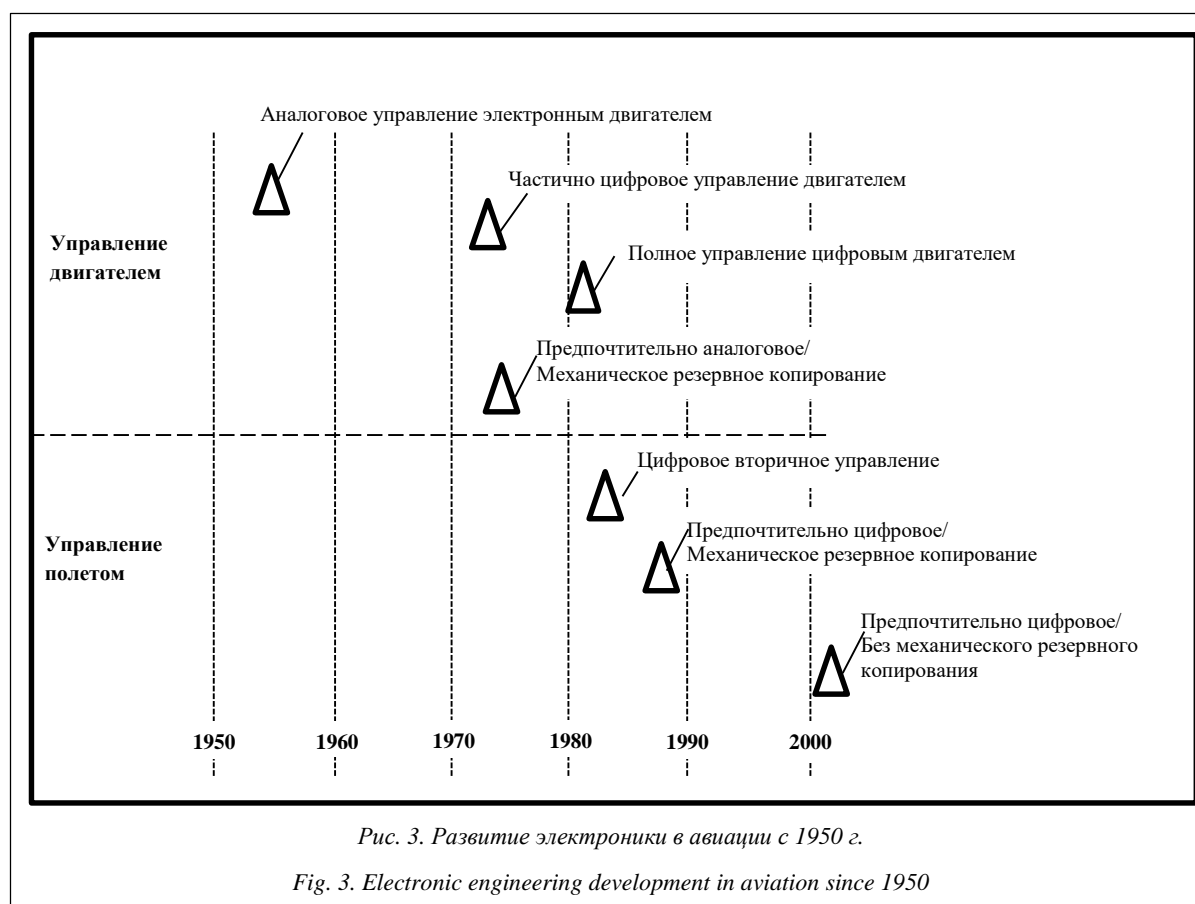
Решающим фактором для применения было наличие экономически выгодных цифровых шин данных, таких как ARINC 429, MIL-STD-1553B и ARINC629. Эта технология в сочетании с дешевыми микропроцессорами и более совершенными инструментами разработки ПО привели к ее широкому применению на борту самолета в общепринятой практике [7].

С появлением понятия загружаемых программных модулей – технологии, применяемой на борту гражданских ВС, ПО больше не воспринимается как часть аппаратных средств, не требующих дополнительных навыков и средств для создания и управления.

С развитием авиационной индустрии число программных модулей, разработанных для систем управления самолетом и корректного функционирования полета, возрастает экспоненциально.

Такие ВС, как Boeing 737 и Airbus320, содержали примерно 30 программных блоков, которые модифицировались довольно редко. С разработкой Boeing 777 в середине 1990-х гг. их число возросло более чем на 120 модулей [8].

Производство таких самолетов, как Boeing 787, увеличило количество ПО до 500 модулей, которые, в свою очередь, загружаются в 800–900 частей аппаратных средств [8].



На сегодняшний день управление ПО ВС является довольно сложной задачей. При неправильно скомпилированном или загруженном в аппаратную часть ПО самолет не может совершать полеты. Поэтому в настоящее время управление ПО и его настройка (конфигурирование), скорее, не опция, а необходимость. Технология загружаемой системы включает в себя как ПО, так и аппаратную часть в виде так называемых загружаемых заменяемых блоков (LRU loadable replacement units), которые независимо друг от друга настраиваются (конфигурируются) на уровне борта самолета. Программные модули могут быть загружены в аппаратные части несколькими способами: с помощью как постоянно установленной программы-загрузчика на борту самолета, так и дополнительного оборудования вне борта.

Большинство современных самолетов, таких как Боинг 737, 747, 767, 777, на борту имеют так называемые загружаемые программные части (loadable software parts).

Авиационные бортовые системы, использующие технологию модульной загрузки ПО, позволяют техническому персоналу, обслуживающему ВС, изменять конфигурацию загружаемых систем без замены аппаратной части. Изменение функциональности системы посредством нового (обновленного) программного блока дает возможность сократить количество заменяемых аппаратных ча-

стей, повысить унификацию аппаратных частей, снизить время обслуживания.

Загружаемое ПО, предназначенное для использования в бортовой аппаратуре ВС гражданского назначения, как правило, разделяется на несколько категорий в соответствии с выполняемыми функциями: операционная система легкозаменяемых аппаратных блоков, настройки действующего ПО, БД, изменяемая информация авиалиний [9].

Операционная система заменяемого аппаратного блока ВС оперирует данными, содержащимися в конфигурационных файлах – файлах настройки для определения функции конкретного аппаратного блока. Операционная система – обычно самый большой и сложный программный модуль как по объему содержащейся в нем информации, так и по времени его загрузки в аппаратную часть.

Файл настройки операционной системы. Этот программный модуль – специализированная БД, которая определяет конфигурацию (настройки) и функцию аппаратного блока, позволяя или запрещающая те или иные опциональные функции, содержащиеся в операционной системе.

Информация о настройках может передаваться по дискретным каналам связи во многие заменяемые аппаратные блоки. Файлы настройки, как правило, требуют для загрузки менее минуты.

БД представляет собой коллекцию данных, классифицированных для простоты использования

и поиска в операционной системе заменяемого аппаратного блока.

Примерами БД, используемых в аппаратных блоках с возможностью загрузки модульного ПО, могут служить БД:

- навигации ЭВМ управления полетом;
- краткого справочника взлетных скоростей ЭВМ управления полетом;
- адресно-отчетной системы авиационной связи;
- системы индикации для общего дисплея.

Навигационная БД – база, содержащая данные о навигации и маршруте полета, которые используются ЭВМ управления полетом для выполнения задач навигации. Как правило, навигационные базы обновляются каждые 28 дней и становятся доступными для загрузки за неделю до их актуализации.

Изменяемая информация авиалинии – это также небольшой файл данных, который предоставляет информацию в операционную систему заменяемых аппаратных блоков. Оператор, загружающий ПО на борт самолета, формирует (генерирует) файл данных информации авиалинии для спецификации таких функций, как формирование отчетов или предоставление сервисов для различных пассажирских зон.

Например, на некоторых ВС операционная система обращается к файлу изменяемой информации авиалиний, когда необходимо сформировать отчет, записать или отформатировать данные.

Файл изменяемой информации авиалиний – файл данных, а не программа или исполняемый файл. Тем не менее данный файл для некоторых систем содержит логические блоки, которые являются программным кодом высокого уровня. Степень изменения/модификации файла изменяемой информации авиалиний контролируется сертифицированной операционной системой заменяемых аппаратных блоков, которая не позволяет вносить оператору изменения, влияющие на безопасность полета, даже если изменения были правильными.

Технологии загрузки информации вышеописанных программных модулей в аппаратную часть борта самолета и ее обмена регламентированы соответствующими стандартами ARINC.

Наравне с методами проектирования и разработки модульной архитектуры ПО в авиационной промышленности создание ПО с применением такой технологии используется в промышленной автоматизации. Примером может служить ПО процессов управления рецептурным производством.

Компания Rockwell Automation предлагает применять термины и рекомендации производственного стандарта ISA 88.01 при разработке ПО и показывает, что для создания модульной архитектуры ПО логически всю систему следует разделить на управляющие модули, модули управления оборудованием, программные модули взаимодействия между ними [10].

Сравнение модульной архитектуры ПО самолета и симулятора

Следует отметить, что к разработке ПО ВС предъявляются более высокие требования, описанные в таких стандартах, как DO178, ARP 4754A. Это направлено на обеспечение безопасности полетов, надежности и качества программных продуктов. К ПО авиационных тренажеров данные стандарты неприменимы либо применимы частично (зависит от уровня тренажера).

Несмотря на отличие требований к разработке ПО гражданского ВС и авиационного тренажера общей является технология построения архитектуры ПО. Модульность присуща как ПО ВС, так и ПО симулятора. При разработке ПО авиационного тренажера предъявляются не такие высокие требования по отказоустойчивости и обеспечению информационного обмена, как при проектировании и разработке ПО ВС.

Обобщая подход к построению модульной архитектуры по авиационной тематике, можно сделать вывод, что для корректного проектирования архитектуры следует сформировать матрицу необходимых программных модулей, распределив их по уровням взаимодействия в зависимости от сложности разрабатываемой системы, применения и типа (типов) информационного обмена.

Логически графический пользовательский интерфейс должен находиться на самом верхнем уровне. Тем не менее, как показывают исследования построения архитектуры ПО авиационных симуляторов [1, 3, 5], управляющие программные модули связаны с графическим пользовательским интерфейсом и находятся на одном уровне. Безусловно, графический пользовательский интерфейс является отдельным программным модулем либо набором программных модулей. Взаимодействие управляющих программных модулей с программными блоками нижних уровней, как и графическое отображение операций, происходящих в системе, а также реакции системы – обратной связи на действия пользователя тесно связаны, поэтому данные программные модули находятся на одном уровне.

Архитектура ПО может быть представлена как матрица размером $M \times N$, где $M = 5$ – число программных уровней, а N – число необходимых программных модулей.

В таблице, которая является графическим представлением проектирования модульной архитектуры ПО на уровне информационного обмена, в качестве примера представлены два наиболее распространенных в авиационных приложениях вида обмена информацией – ARINC 429 и MIL-STD 1553. Следовательно, для корректного функционирования данной системы необходимо разработать два программных модуля обмена, логически соответствующих выбранным стандартам.

Графическое представление матрицы формирования архитектуры ПО

Graphical presentation of the software design architecture matrix

Уровень	Программный блок		
	1	2	N
1. Управляющие программные блоки, человеко-машинное взаимодействие	Main	Графический пользовательский интерфейс (конечное число программных блоков зависит от среды разработки и целевой платформы для реализации)	
2. Начальные условия, оперирование переменными	Блок переменных 1		
3. Информационный обмен	ARINC 429	MIL-STD 1553	
4. Математические модели самолета, реализация специфических алгоритмов для конкретной модели/типа самолета	Мат. модель 1	Мат. модель 2	Мат. модель 3
5. Массивы данных, таблицы, константы, необходимые для вычислений	Таблица 1		

Например, при разработке системного ПО уровень графического пользовательского интерфейса не применяется, но управляющие программные модули все равно присутствуют. Поэтому при проектировании такого типа ПО также необходимо оставить пять уровней, где модуль графического пользовательского интерфейса будет отсутствовать, а основные управляющие модули (например main) – нет. То есть его архитектура может быть представлена как матрица размером $(M-1) \times N = 4 \times N$, где N – число необходимых программных модулей.

Также при проектировании сложных систем общая матрица – архитектура сложной программной системы может включать составные блоки. Например, логически связанные блоки, взаимодействующие с аппаратной средой и имеющие свои алгоритмы и циклограмму выполнения задач, можно интегрировать в систему, имеющую свой графический пользовательский интерфейс, посредством внесения изменений в программные блоки либо создания новых программных модулей информационного обмена (уровень 3), переменных (уровень 2) и массивов данных (уровень 5).

Выводы

Как видно из приведенных примеров, при проектировании ПО гражданских ВС и авиационных тренажеров наиболее удобным и эффективным подходом является создание ПО авиационного назначения с применением модульной архитектуры. Главным преимуществом применения данной технологии в бортовом ПО ВС является изменение функциональности без изменения аппаратной части, что позволяет сократить количество заменяемых аппаратных частей, которые должны хранить как компании, обслуживающие ВС, так и компании-производители ВС.

Можно также выделить следующие преимущества при разработке ПО с использованием модульной архитектуры построения для разработчиков ПО.

- Снижение времени на конструирование (проектирование) архитектуры новых программных продуктов. Используя при создании ПО данную технологию и при необходимости производственный стандарт ISA 88.01, разработчики могут сконцентрироваться на функциональности управляющих программных модулей, специфических для конкретной машины, а не на технике реализации, тем самым сокращая время разработки. Данный подход к разработке также может способствовать синхронной разработке программных блоков группой разработчиков, что, в свою очередь, дополнительно ускорит выполнение каждого цикла разработки и, как результат, сократит общее время на разработку программной системы.

- Снижение экономических затрат на проектирование (конструирование), разработку, поставку (в том числе развертывание) программных продуктов. В дополнение к сокращению времени на разработку использование технологии модульной архитектуры и разработки ПО также помогает снизить затраты в течение всего его жизненного цикла, включающего конструирование (разработку) архитектуры, программирование, тестирование, введение в эксплуатацию, сопровождение.

- Возможность вносить больше инноваций в машиностроение, увеличивать производительность создаваемой техники.

Преимуществами использования разработчиками ПО технологии модульной архитектуры ПО для производителей аппаратной части оборудования являются следующие:

- гибкость процесса производства: возможность адаптировать существующие ресурсы к нуж-

дам и требованиям нового продукта с минимальными экономическими затратами и затратами времени;

– улучшение существующих ресурсов и повышение производительности: возможность упростить непрерывное усовершенствование путем вертикальной интеграции существующих ресурсов оборудования;

– безопасность и соответствие нормативным правилам: модульное программирование упрощает соблюдение нормативных и сертификационных требований, а также проверку за счет сокращения объема усилий по проверке только модифицированных модулей для аппаратной части, а не целых программных приложений [10].

Данный подход позволяет снизить трудозатраты на обслуживание уже разработанного ПО посредством замены или доработки программного модуля, не изменяя при этом всю структуру системы, повысить эффективность и качество разрабатываемого ПО, так как при модульном подходе к разработке снижается вероятность появления ошибок, которые сложно отследить в программном коде. Метод проектирования ПО с помощью модульной архитектуры стимулирует и облегчает непрерывное совершенствование программных продуктов, снижая риск отрицательного эффекта на всю программную систему в целом.

Преимуществами применения модульной архитектуры также является возможность применения при наукоемких вычислениях и решении задач параллельных вычислений.

Таким образом, при проектировании сложных систем необходимо учитывать и проектировать интерфейсы информационного взаимодействия, вносить необходимые дополнения с учетом имею-

щихся программных модулей для корректного функционирования всей системы в целом и располагать как программные модули взаимодействия, так и проектируемые программные модули в соответствии с предложенной матрицей.

Литература

1. Hoogstraten J.A., van de Moesdijk G.A.J. Modular programming structure applied to the simulation of non-linear aircraft models. URL: <http://repository.tudelft.nl/islandora/object/uuid%3A0a0508b9-7e2a-4f61-b3ce-006644642937> (дата обращения: 20.09.2016).
2. Данилов А.М., Лапшин Э.В., Гушина А.А. Информационно-вычислительные системы авиационных тренажеров модульной архитектуры с распараллеливанием вычислительных процессов // Надежность и качество: тр. Междунар. симпози. Пенза: Изд-во Пензенск. гос. ун-та, 2006. Т. 1. С. 39–44.
3. Baarspul M. Flight Simulation Techniques with emphasis on the generation of high fidelity 6 DOF motion cues. URL: <http://repository.tudelft.nl/islandora/object/uuid%3Adfd19219-afd9-4faf-b2c8-03401bfc89c0?collection=research> (дата обращения: 20.09.2016).
4. Данилов А.М., Лапшин Э.В. Теория и практика имитационного моделирования и создания тренажеров // Приборы и системы управления. 1989. № 8. С. 11–12.
5. Baarspul M. Lecture Notes on Flight Simulation Techniques Report LR-596. 1989. URL: <http://resolver.tudelft.nl/uuid:269fa44e-0b09-4fc5-9260-83b6afb7cdd> (дата обращения: 20.09.2016).
6. Lovell B. Echoes of war: the story of H2S radar. Adam Hilger Ltd., Bristol, 1991, 287 p.
7. Moir I., Seabridge A. Design and development of Aircraft Systems. UK, Wiley, 2012, 334 p.
8. Best practices for loadable software management and configuration control. Documents, IATA – Eng. and Maintenance Group, Montreal, Canada, 2013, 11 p.
9. Klein J. Onboard Loadable Software Boeing. URL: http://www.boeing.com/commercial/aeromagazine/aero_05/ps/ps02/index.html (дата обращения: 20.09.2016).
10. Integrated Architecture™: foundations of modular programming. Ref. Document, Rockwell Automation, Inc. Publ., 2010, 86 p.

Software & Systems

DOI: 10.15827/0236-235X.118.291-300

Received 10.11.16

2017, vol. 30, no. 2, pp. 291–300

DESIGN PRINCIPLES OF MODULAR SOFTWARE ARCHITECTURE IN AVIATION

L.A. Chizhikova¹, *Leading Engineer, l_chizhikova@scac.ru*

¹ JSC “Sukhoi Civil Aircraft”, Polikarpova St. 23b, bldg. 2, Moscow, 125284, Russian Federation

Abstract. Software development is a quite difficult and laborious process, which is based on correct and reliable architecture design. Distribution and coordination of duties when working in a group of developers is often a quite difficult and responsible decision due to affection the main result of development.

Nowadays, with technological development the control and functionality of aircraft avionics requires more software creation. The role of software development and management is also increasing in the flight simulation and testing equipment fields. More and more tasks are transferred from hardware to software.

The article provides the analysis and identification of basic aviation software design aspects, the comparison between the civil aircraft software design architecture and flight simulation software architecture.

In order to present a unified software architecture model, the authors made a research on papers on software architecture design in Russian companies, which make Flight Simulation devices and onboard aircraft software architecture. There is also a comparison of these solutions and their commonality. Furthermore, the article considers a model, which is successfully used

in software development by Rockwell Automation Company, and makes a research among the works of Delft Technical University on the described topic.

The paper proposes a mathematical description of modular software product architecture, which is oriented to the aviation industry.

The proposed approach allows unifying a development process, reducing the time and labor of software development, adding innovations without existing structure transformation if the software product is made using the described solution.

Keywords: software, software architecture, software architecture design, software development, flight simulation, aviation software, mathematical modeling.

References

1. Hoogstraten J.A., van de Moesdijk G.A.J. Modular programming structure applied to the simulation of non-linear aircraft models. *IMACS Symp. Simulation in Engineering Sciences*. Delft Univ. of Technology, Nantes, 1983, 11 p. Available at: <http://repository.tudelft.nl/islandora/object/uuid%3A0a0508b9-7e2a-4f61-b3ce-006644642937> (accessed September 20, 2016).
2. Danilov A.M., Lapshin E.V., Gushchina A.A. Information and computational systems of modular architecture flight simulation devices with parallel computational processes. *Trudy mezhdunar. simp. "Nadezhnost i kachestvo"* [Proc. Int. Symp. "Quality and Reliability"]. Vol. 1, Penzensky Gos. Univ. Publ., Penza, 2006, pp. 39–44 (in Russ.).
3. Baarspul M. Flight Simulation Techniques with emphasis on the generation of high fidelity 6 DOF motion cues. *Proc. 15th Congress of the Int. Council of the Aeronautical Sciences*. Delft Univ. of Technology Publ., London, 1986, 17 p. Available at: <http://repository.tudelft.nl/islandora/object/uuid%3Adfd19219-afd9-4faf-b2c8-03401bfc89c0?collection=research> (accessed September 20, 2016).
4. Danilov A.M., Lapshin E.V. The theory and practice of simulation modeling and flight simulation creation. *Pribory i sistemy upravleniya* [Instruments and Control Systems]. 1989, no. 8, pp. 11–12 (in Russ.).
5. Baarspul M. *Lecture Notes on Flight Simulation Techniques Report LR-596*. Delft Univ. of Technology Publ., 1989. Available at: <http://resolver.tudelft.nl/uuid:269fa44e-0b09-4fc5-9260-83b6affb7cdd> (accessed September 20, 2016).
6. Lovell B. *Echoes of War: The Story of H2S Radar*. Adam Hilger (Bristol) Publ., 1991, 287 p.
7. Moir I., Seabridge A. *Design and development of Aircraft Systems*. UK, Wiley Publ., 2012, 334 p.
8. *Best Practices for Loadable Software Management and Configuration Control*. IATA Int. Air Transport Association Engineering and Maintenance Group, Montreal, 2013, 11 p.
9. Klein J. Onboard Loadable Software Boeing. *AERO magazine*. 1998, no. 5. Available at: http://www.boeing.com/commercial/aeromagazine/aero_05/ps/ps02/index.html (accessed September 20, 2016).
10. Bradley A. *Integrated Architecture Foundations of Modular Programming*. Rockwell Automation. Rockwell Software Publ., 2010, 86 p.

Примеры библиографического описания статьи

1. Чижикова Л.А. Принципы проектирования модульной архитектуры программного обеспечения авиационной тематики // Программные продукты и системы. 2017. Т. 30. № 2. С. 291–300; DOI: 10.15827/0236-235X.118.291-300.
2. Chizhikova L.A. Design principles of modular software architecture in aviation *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 291–300 (in Russ.); DOI: 10.15827/0236-235X.118.291-300.

УДК 621.391

DOI: 10.15827/0236-235X.118.301-306

Дата подачи статьи: 06.04.17

2017. Т. 30. № 2. С. 301–306

УНИФИЦИРОВАННОЕ ОПИСАНИЕ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННЫХ РАДИОЭЛЕКТРОННЫХ СИСТЕМ ДЛЯ ОЦЕНКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УЧЕБНО-ТРЕНИРОВОЧНЫХ СРЕДСТВ

Ф.Н. Абу-Абед, к.т.н., доцент, aafares@mail.ru

(Тверской государственный технический университет,
наб. Аф. Никитина, 22, г. Тверь, 170026, Россия);

Р.В. Допира, д.т.н., профессор, зав. отделом, rvdopira@yandex.ru

(Научно-производственное объединение «Русские базовые информационные технологии»,
просп. Калинина, 17, г. Тверь, 170001, Россия);

И.Н. Ищук, д.т.н., доцент, начальник кафедры, Boerby@rambler.ru;

Е.Ю. Брежнев, к.т.н., докторант, dimanbreg@mail.ru;

А.Н. Потанов, к.т.н., доцент, зам. начальника кафедры, potanov_il@mail.ru

(Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия»
им. проф. Н.Е. Жуковского и Ю.А. Гагарина,
ул. Старых Большевиков, 54а, г. Воронеж, 394064, Россия)

В настоящее время оценка ПО учебно-тренировочных средств выполняется по степени совпадения моделируемых процессов с реальными. Однако отсутствие универсальности описания механизмов функционирования эрготехнических информационных радиоэлектронных систем, для которых разрабатываются тренажные комплексы, не обеспечивает выработку инструментария для формирования стратегий их применения с учетом изменения условий функционирования, инвариантного к радиоэлектронным системам. Вследствие этого актуальным является унифицированное описание (представление) функционирования информационных радиоэлектронных систем, на основе которого предполагается создавать ПО их учебно-тренировочных средств.

С учетом этого целью работы является унифицированное описание функционирования информационных радиоэлектронных систем, на основе которого в дальнейшем могла быть реализована оценка ПО их учебно-тренировочных средств. Унифицированное описание функционирования информационных радиоэлектронных систем основывается на едином подходе к их представлению в виде структурно-логических схем, отличительной особенностью которых является то, что в них учитываются радиочастотные (РЧ) взаимодействия с элементами среды, выраженные через РЧ-действия и РЧ-влияния, образующие причинно-следственные отношения РЧ-условий функционирования РЭС. Возможности осуществления радиоэлектронных систем функций выбора, распределения и перераспределения ресурсов зависят от заложенных в нем механизмов управляемости. Для построения структурно-логических схем функционирования эрготехнических радиоэлектронных систем, учитывающих операторскую деятельность, в работе отдельно рассмотрены механизмы их управляемости.

В дальнейшем это позволяет разработать научно-методический аппарат оценки ПО учебно-тренировочных средств операторов управления воздушным движением по результатам экспертизы, а также, при представлении их как систем сложной иерархической структуры построения, по информационному полю рабочих мест операторов управления воздушным движением с использованием критерия Фишера- χ^2 и обобщенного закона распределения.

Ключевые слова: эрготехнический, радиоэлектронная система, структурно-логический, управление, модель, автоматизированный.

Принимая во внимание то, что по своему функциональному назначению все радиоэлектронные системы (РЭС) основаны на использовании радиоволн, в работах [1, 2] определена классификация самих РЭС, которая подтверждает справедливость их рассмотрения относительно собственно радиочастотных (РЧ) действий. Это позволяет сформировать единый универсальный подход к представлению РЭС, инвариантный к РЧ-условиям. Выработка рекомендаций по применению РЭС S должна учитывать как общность действий в РЧ-спектре, так и возможности их реализации.

Унифицированное представление функционирования информационных РЭС

Под общностью R_f РЧ-спектра будем понимать непустое множество радиочастот $R_f = \{f_k\}$, образу-

ющееся в результате пересечения множеств радиочастот $f_Q = \{f_{Qj}\}$ и $f_S = \{f_{Sp}\}$, используемых, соответственно, РЭС S и средой Q :

$$R_f: f_Q = f_Q \cap f_S \neq \emptyset. \quad (1)$$

Элемент f_k множества радиочастот R_f образуется в результате

$$f_k: (f_{Qj} \wedge f_{Si} = f_{Qj}) \vee (f_{Qj} \wedge f_{Si} = f_{Si}). \quad (2)$$

Отсутствие общности $\bar{R}_f = \{\bar{f}_k\}$ РЧ-спектра наблюдается, когда $\bar{R}_f: f_Q \cap f_S = \emptyset$, то есть

$$\bar{f}_k: (f_{Qj} \wedge f_{Si} = 0) \vee (f_{Qj} \wedge f_{Si} = 0).$$

Схематично прямую РЧ-связанность d_f^+ РЭС S

со средой Q можно представить в виде $d_f^+: S \xrightarrow{R_f} Q$.

Аналогичным образом можно представить об-

ратную РЧ-связанность d_f^- РЭС S со средой Q :

$$d_f^- : S \xleftarrow{R_f} Q.$$

Если существуют одновременно d_f^+ и d_f^- , значит, РЭС S и среда Q являются взаимосвязанными в РЧ-спектре:

$$d_f^\pm = d_f^+ \wedge d_f^- = (S \xrightarrow{R_f} Q) \wedge (S \xleftarrow{R_f} Q) = S \leftrightarrow Q. \quad (3)$$

Тогда можно говорить об отношениях

– $\bar{d}_f^+ : S \xrightarrow{\bar{R}_f} Q$ – объект S не имеет прямую РЧ-связанность со средой Q ;

– $\bar{d}_f^- : S \xleftarrow{\bar{R}_f} Q$ – объект S не имеет обратную РЧ-связанность со средой Q ;

– $\bar{d}_f^\pm : S \leftrightarrow Q$ – объект S не имеет как прямую, так и обратную РЧ-связанность со средой Q .

В соответствии с этим образуются $d_f = \{d_f^+, d_f^-, d_f^\pm\} = \{d_f^\lambda\}$ и $\bar{d}_f = \{\bar{d}_f^+, \bar{d}_f^-, \bar{d}_f^\pm\} = \{\bar{d}_f^\lambda\}$ – множества отношений, соответственно, РЧ-связанности и несвязанности ($\lambda = +, -, \pm$).

Так как между любым элементом Q_j среды Q и РЭС S существуют либо отношения $d_f^\lambda \in d_f$, либо отношения $\bar{d}_f^\lambda \in \bar{d}_f$, с учетом введенных понятий среда Q может состоять из множества РЧ-связанных Q_C и множества несвязанных Q_D элементов, то есть

$$Q_C = \{Q_{Cj} : (Q_j, S) \notin \bar{d}_f\} = \{Q_{Cj} : (Q_j, S) \in d_f\}, \quad (4)$$

$$Q_D = \{Q_{Dj} : (Q_j, S) \notin d_f\} = \{Q_{Dj} : (Q_j, S) \in \bar{d}_f\}, \quad (5)$$

причем

$$Q = \{Q_{Cj} : (Q_j, S) \in d_f\} \cup \{Q_{Dj} : (Q_j, S) \in \bar{d}_f\}. \quad (6)$$

Особый интерес вызывает множество РЧ-связанности Q_C элементов среды Q с РЭС S , так как, если $Q_C = \{Q_{Cj}\} = \emptyset$, функционирование последнего в интересах достижения желаемого результата P_S теряет свой смысл. Поэтому $Q_C \neq \emptyset$ – *первый сопутствующий признак* функционирования РЭС S .

Естественно, если $Q_C \neq \emptyset$, а $\Delta_S = \emptyset$, $\Delta_S \subseteq S$ и $\Delta_Q = \emptyset$, $\Delta_Q \subseteq Q$, либо $\Delta_S \neq \emptyset$ и $\Delta_Q = \emptyset$, либо $\Delta_S = \emptyset$ и $\Delta_Q \neq \emptyset$, то функционирование РЭС S в целях достижения желаемого результата P_S также теряет свой смысл. Поэтому $\Delta_S \neq \emptyset$ и $\Delta_Q \neq \emptyset$ – *второй сопутствующий признак* функционирования РЭС S .

Очевидно, если $\Delta_S \neq \emptyset$ достигают $\Delta_Q \neq \emptyset$, а это возможно тогда и только тогда, когда $Q_C \neq \emptyset$, то возникает множество воздействий $\pi_{SQ} = \{\pi_{SQn}\}$, соответственно, РЭС S на среду Q :

$$\pi_{SQ} = \Delta_S \cap \Delta_Q, \pi_{SQ} \subseteq Q_C.$$

Поэтому $\pi_{SQ} \neq \emptyset$ – *третий сопутствующий признак* функционирования S .

Варианты формирования воздействия π_{SQ} РЭС S на среду Q могут быть различными:

- прямое воздействие $\pi_{SQ}^+ : (\Delta_S \xrightarrow{Q_C} \Delta_Q)$;
- обратное воздействие $\pi_{SQ}^- : (\Delta_S \xleftarrow{Q_C} \Delta_Q)$;
- взаимное воздействие $\pi_{SQ}^\pm : (\Delta_S \leftrightarrow^{Q_C} \Delta_Q)$.

Воздействия π_{SQ} РЭС S на среду Q по природе формирования эквивалентны воздействиям π_{SQ} среды Q на РЭС S , то есть

$$\begin{aligned} \pi_{SQ}^+ &\sim \pi_{QS}^- : (\Delta_Q \xleftarrow{Q_C} \Delta_S), \\ \pi_{SQ}^- &\sim \pi_{QS}^+ : (\Delta_Q \xrightarrow{Q_C} \Delta_S), \\ \pi_{SQ}^\pm &\sim \pi_{QS}^\pm : (\Delta_Q \leftrightarrow^{Q_C} \Delta_S). \end{aligned} \quad (7)$$

Если условно принять, что $Q_C \neq \emptyset$ и $\Delta_Q \neq \emptyset$ являются влиянием β_Q среды Q на трансформацию действий Δ_S РЭС S в воздействия π_{SQ} , то есть

$$\begin{aligned} \beta_Q : ((Q_C \neq \emptyset) \vee (\Delta_Q \neq \emptyset)), \\ \text{то справедлива следующая запись:} \\ \Delta_S \rightarrow \beta_Q \rightarrow \pi_{SQ}. \end{aligned} \quad (8)$$

Аналогичным образом получаем логическую цепочку по формированию воздействия среды Q на РЭС S :

$$\Delta_Q \rightarrow \beta_S \rightarrow \pi_{SQ}, \quad (9)$$

где $\beta_S : ((Q_C \neq \emptyset) \vee (\Delta_S \neq \emptyset))$.

Эти логические цепочки наглядно представляют процесс формирования воздействий π_{SQ} и π_{QS} в РЧ-спектре.

С учетом вышеизложенного унифицированное представление функционирования активно-пассивных РЭС S описывается в следующем виде [3, 4]:

– первично РЭС формирует в РЧ-спектре действие Δ_S , направленное на излучение радиоволн y_S ($\Delta_S: y_S$);

– в случае наличия в РЧ-спектре влияния β_Q среды Q часть действия $\Delta_S: y_S$ трансформируется в воздействие π_{SQ} , радиоволны y'_S которого являются радиоволнами x'_Q , принимаемыми средой Q , то есть $\pi_{SQ} : y'_S = x'_Q$;

– в соответствии с принимаемыми радиоволнами x'_Q среда Q осуществляет излучение собственных радиоволн y_Q , которые являются ее реакцией $\text{Re}(Q)$ на воздействие π_{SQ} РЭС S , то есть $y_Q = \text{Re}(Q) : \pi_{SQ}$;

– излучение средой Q радиоволн y_Q характеризует в РЧ-спектре ее действие Δ_Q , то есть $\Delta_Q: y_Q$;

– в случае наличия в РЧ-спектре влияния β_S объекта S часть действия $\Delta_Q: y_Q$ трансформируется в воздействие π_{QS} , радиоволны y'_Q которого явля-

ются радиоволнами x'_s , принимаемыми РЭС S , то есть $\pi_{qs} : (y'_q = x'_s)$.

Это можно отобразить как

$$\Delta_s: y_s \rightarrow \beta_q \rightarrow \pi_{sq} : (y'_s = x'_q) \rightarrow \langle y_q = \text{Re}(Q) : \pi_{sq} \rangle \rightarrow \Delta_q: y_q \rightarrow \beta_s \rightarrow \pi_{qs} : (y'_q = x'_s). \quad (10)$$

Процесс унифицированного представления [5, 6]

– для пассивно-активных РЭС:

$$\Delta_q: y_q \rightarrow \beta_s \rightarrow \pi_{qs} : (y'_q = x'_s) \rightarrow \langle y_s = \text{Re}(S) : \pi_{qs} \rangle \rightarrow \Delta_s: y_s \rightarrow \beta_q \rightarrow \pi_{sq} : y'_s = x'_q; \quad (11)$$

– для активно-адаптивных РЭС:

$$\Delta_s: y_s \rightarrow \beta_q \rightarrow \pi_{sq} : (y'_s = x'_q) \rightarrow \langle y_q = \text{Re}(Q) : \pi_{sq} \rangle \rightarrow \Delta_q: y_q \rightarrow \beta_s \rightarrow \pi_{qs} : (y'_q = x'_s) \rightarrow \langle y_s = \text{Re}(S) : \pi_{qs} \rangle; \quad (12)$$

– для пассивно-адаптивных РЭС:

$$\Delta_q: y_q \rightarrow \beta_s \rightarrow \pi_{qs} : (y'_q = x'_s) \rightarrow \langle y_s = \text{Re}(S) : \pi_{qs} \rangle \rightarrow \Delta_s: y_s \rightarrow \beta_q \rightarrow \pi_{sq} : (y'_s = x'_q) \rightarrow \langle y_q = \text{Re}(Q) : \pi_{sq} \rangle, \quad (13)$$

где $\text{Re}(Q)$ – реакция среды Q ; $x'_s \subseteq x_s$, $y'_s \subseteq y_s$, $x'_q \subseteq x_q$ и $y'_q \subseteq y_q$.

В этих структурно-логических схемах в явном виде отсутствует учет ресурсов РЭС, а значит, нет возможности их использования в представленном виде для определения содержательных компонентов учебных упражнений. В [7] рассмотрен пример, из которого видно, что эффективность применения РЭС зависит не только от имеющихся ресурсов, в частности энергетических, но и от возможности реализации объектом функций их перераспределения, в частности излучаемой мощности.

Механизмы управляемости

Возможность осуществления РЭС функций выбора, распределения и перераспределения ресурсов зависит от заложенных в нем механизмов управляемости. Для построения структурно-логических схем функционирования эрготехнических РЭС, учитывающих операторскую деятельность, рассмотрим отдельно механизмы их управляемости.

Как определено в [3], формирование РЧ-воздействия π_{sq} зависит от множества располагаемых РЭС S ресурсов [6, 8]:

$$R_s = \{f_s, t_s, W_s, \xi_s\}, \quad (14)$$

где f_s – РЧ-ресурсы (первообразные множества РЧ-связанности $Q_c = \{Q_c\} \neq \emptyset$); t_s , W_s и ξ_s – соответственно временные, энергетические и пространственные ресурсы (первообразные множества действия $\Delta_s \neq \emptyset$).

В свою очередь, располагаемые ресурсы R_s РЭС S могут характеризоваться следующими параметрами [4]:

– f_s – рабочая радиочастота f_{s0} , РЧ-диапазон Δf_s , полоса пропускания ΔF_s и т.п.;

– t_s – момент времени t_{s0} возникновения действия Δ_s , длительность τ_s , периодичность T_s действия Δ_s и т.п.;

– W_s – энергия E_s , затрачиваемая на формирование действия Δ_s , средняя мощность p_s , импульсная мощность p_{sl} , коэффициент полезного действия η_s и т.п.;

– ξ_s – ширина распространения (проникновения) $\Delta\theta_s$ и $\Delta\Phi_s$ действия Δ_s , соответственно, в азимутальной и угломестной плоскости, азимут θ_s и угол места Φ_s распространения максимума действия Δ_s , пространственная поляризация γ_{Δ_s} действия Δ_s и т.п.

Сами располагаемые ресурсы R_s определяют потенциальные возможности применения РЭС S .

Известно, что в РЧ-спектре РЭС S при воздействии π_{qs} на него среды Q может формировать реакцию $\text{Re}(S)$ в виде ответных действий:

$$\Delta_s = \text{Re}(S) : \pi_{qs}. \quad (15)$$

Реакция $\text{Re}(S)$ может быть управляемой и неуправляемой.

В свою очередь, РЭС S можно представить в виде

$$S \subseteq \pi_{qs} \times \Delta_s, \Delta_s = \text{Re}(S) : (\pi_{qs}, C_s) \quad (16)$$

$$\text{или } \pi_{qs} \rightarrow C_s \rightarrow \Delta_s, \quad (17)$$

где C_s – множество состояний S [4].

Если $C_s \neq C_s(\pi_{qs})$, то S является условно управляемым в РЧ-спектре, то есть его управление заключается в адаптации действий Δ_s на воздействия π_{qs} без изменения C_s :

$$\Delta_s \rightarrow \beta_q \rightarrow \pi_{sq} \rightarrow \Delta_q = \text{Re}(Q) : \pi_{sq} \rightarrow \beta_s \rightarrow \pi_{qs} \rightarrow C_s. \quad (18)$$

Если $C_s = C_s(\pi_{qs})$, то S является безусловно управляемым в РЧ-спектре, то есть его управление заключается в изменении состояния C_s на воздействия π_{qs} :

$$\pi_{sq} \rightarrow C_s(\pi_{sq}) \rightarrow \Delta_s. \quad (19)$$

Объект S может быть комплексно управляемым в РЧ-спектре, то есть для одной части элементов множества $C_s = C_s(\pi_{qs})$, а другой части – $C_s \neq C_s(\pi_{qs})$:

$$\Delta_s \rightarrow \beta_q \rightarrow \pi_{sq} \rightarrow \Delta_q = \text{Re}(Q) : \pi_{sq} \rightarrow \beta_s \rightarrow \pi_{qs} \rightarrow C_s(\pi_{qs}).$$

Необходимо отметить, что, помимо внешней управляемости (по радиоканалам) РЭС S , он может быть внутренне управляемым (по электрическим цепям). Внутренняя управляемость состоит в изменении состояния C_s РЭС S в зависимости от текущей полезности q_s его функционирования и желаемого результата P_s для сложившихся РЧ-условий.

Полезность функционирования РЭС S [3] представим в виде

$$q_s = q_s(\pi_{qs}, \Delta_s). \quad (20)$$

Если текущая величина функции полезности РЭС S отличается от желаемого результата P_s , то необходимо формировать такое управляющее воздействие δ_s на состояние C_s , при котором $q_s \rightarrow P_s$, то есть

$$\delta_s(C_s): q_s \rightarrow P_s \quad (21)$$

$$\text{или } \delta_s = F(q_s - P_s), \quad (22)$$

где F – функционал.

В свою очередь, состояние C_s зависит от располагаемых ресурсов R_s [1]:

$$C_s = \text{Re}(S): R_s. \quad (23)$$

Поэтому управляющее воздействие δ_s изначально заключается в выборе и распределении (перераспределении) располагаемого ресурса R_s .

С учетом этого процедура внутреннего управления РЭС S , направленная на формирование в РЧ-спектре его действия Δ_s в соответствии с воздействиями π_{qs} среды Q , может быть представлена как

$$\pi_{qs} \rightarrow \delta_s = F(q_s - P_s) \rightarrow R_s \rightarrow C_s \rightarrow \Delta_s. \quad (24)$$

Необходимо отметить, что сама процедура внутреннего управления РЭС S может быть организованной и неорганизованной [4].

Организационное управление РЭС S_o заключается в выборе и распределении (перераспределении) непосредственно человеком (оператором) располагаемого ресурса R_s , направленных на минимизацию невязки между действительной полезностью q_s и желаемым результатом P_s функционирования объекта и реализуемых оператором с помощью органов управления.

Неорганизованное управление РЭС S_{no} – это управление располагаемыми ресурсами R_s , выполняемое без участия человека (оператора) на основании собственных действий, которые по отношению к объекту являются внутренними.

Ресурсы R_s обеспечивают формирование как РЧ-действий Δ_s , так и информационных потоков Ξ , на основании анализа которых оператор осуществляет операции D по воздействию на органы управления объекта S [5].

С учетом этого функционирование *адаптивно-пассивной* эрготехнической РЭС S представим в следующем виде [5]:

– для автоматизированного режима ($\delta_s \neq \emptyset$, $U_s \neq \emptyset$):

$$\begin{aligned} \pi_{qs} \rightarrow \delta_s = F(q_s - P_s) \vee (\wedge) \Xi \rightarrow \\ \rightarrow \langle \text{Оператор} : \Psi_s \rangle \rightarrow D \rightarrow U_D \rightarrow \\ \rightarrow \vee (\wedge) R_s \rightarrow C_s \rightarrow \Delta_s \rightarrow \beta_{sq} \rightarrow \pi_{sq} \rightarrow \\ \rightarrow \langle \Delta_q = \text{Re}(Q) : \pi_{sq} \rangle \rightarrow \beta_{qs} \rightarrow \pi_{qs}; \end{aligned} \quad (25)$$

– для ручного режима ($\delta_s = \emptyset$, $U_s \neq \emptyset$):

$$\begin{aligned} \pi_{qs} \rightarrow q_s \rightarrow \Xi \rightarrow \langle \text{Оператор} : \Psi_s \rangle \rightarrow \\ \rightarrow D \rightarrow U_D \rightarrow R_s \rightarrow C_s \rightarrow \Delta_s \rightarrow \beta_{sq} \rightarrow \\ \rightarrow \pi_{sq} \rightarrow \langle \Delta_q = \text{Re}(Q) : \pi_{sq} \rangle \rightarrow \beta_{qs} \rightarrow \pi_{qs}. \end{aligned} \quad (26)$$

Соответственно функционирование автоматического РЭС S ($\delta_s \neq \emptyset$, $U_s = \emptyset$) имеет следующее представление:

$$\begin{aligned} \pi_{qs} \rightarrow \delta_s = F(q_s - P_s) \rightarrow R_s \rightarrow \\ \rightarrow C_s \rightarrow \Delta_s \rightarrow \beta_{sq} \rightarrow \pi_{sq} \rightarrow \langle \Delta_q = \\ = \text{Re}(Q) : \pi_{sq} \rangle \rightarrow \beta_{qs} \rightarrow \pi_{qs}. \end{aligned} \quad (27)$$

Структурно-логические модели (25) и (26) определяют прямую задачу по регулированию РЧ-условий функционирования РЭС за счет выполнения операторами предписанных операций D в соответствии с ранее сформированными у них перцептивными образами.

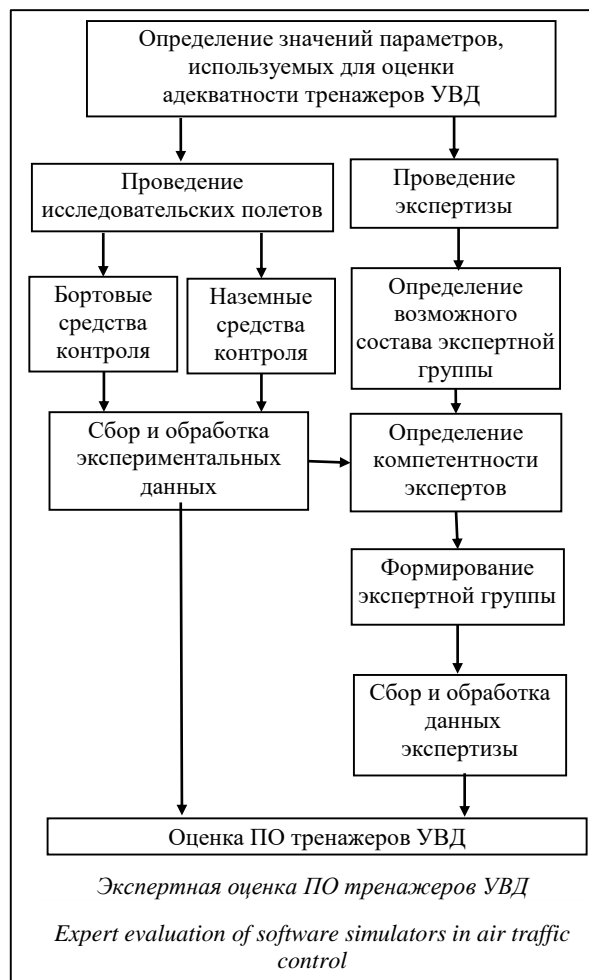
Использование системного подхода и теоретико-множественного описания систем позволило на основании общих для всех РЭС РЧ-признаков классифицировать и определить единый подход представления РЭС в виде структурно-логических схем, отличительной особенностью которых является то, что в них учитываются РЧ-взаимодействия с элементами среды, выраженные через РЧ-действия и РЧ-влияния, образующие причинно-следственные отношения РЧ-условий функционирования РЭС. На основании единого подхода к представлению РЭС, а также учитывая их управляемость, уточнено структурно-логическое описание унифицированного представления функционирования информационных РЭС [9, 10].

Таким образом, унифицированное описание функционирования информационных РЭС основано на едином подходе к их представлению в виде структурно-логических схем, отличительной особенностью которых является то, что в них учитываются РЧ-взаимодействия с элементами среды, выраженные через РЧ-действия и РЧ-влияния, образующие причинно-следственные отношения РЧ-условий функционирования РЭС.

Оценка ПО учебно-тренировочных средств

Данная оценка при унифицированном описании функционирования информационных РЭС может быть выполнена по результатам экспертизы, а также при представлении их как систем сложной иерархической структуры построения и по информационному полю рабочих мест операторов *управления воздушным движением* (УВД) с использованием критерия Фишера- χ^2 и обобщенного закона распределения.

На рисунке представлена структура экспертной оценки имитационного моделирования учебно-тренировочных комплексов операторов УВД.



На рисунке (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/1.jpg) представлен интерфейс программы оценки ПО (оценки адекватности) учебно-тренировочных комплексов операторов УВД и управления планированием тренажной подготовки специалистов по эксплуатации информационных РЭС.

Рассмотренное унифицированное описание функционирования информационных РЭС реализуемо для учебно-тренировочных средств операторов

УВД. Это позволяет в дальнейшем выполнить оценку результатов имитационного моделирования в учебно-тренировочных средствах операторов УВД и скорректировать организацию тренажной подготовки.

Литература

1. Потапов А.Н. Автоматизация тренажной подготовки операторов радиоэлектронных объектов управления воздушным движением. Воронеж: Изд-во ВАИУ, 2010. 136 с.
2. Сысоев Е.С. Особенности функционирования эргатических радиоэлектронных средств управления воздушным движением // Научные чтения имени А.С. Попова: сб. стат. регион. науч.-практич. конф. Воронеж: Изд-во ВУНЦ ВВС «ВВА», 2012. С. 38–42.
3. Лебедев В.В., Романов А.В. Теоретико-множественный подход к описанию функционирования эрготехнических радиоэлектронных средств // Проблемы эффективности и безопасности функционирования сложных технических и информационных систем: сб. тр. 32 Всерос. науч.-технич. конф. Серпухов: Изд-во филиала ВА РВСН им. Петра Великого, 2013. С. 118–122.
4. Сысоев В.В., Крутских П.П., Свинцов А.А. Математическая модель информационного конфликта // Радиотехника (журнал в журнале). 1999. № 3. С. 77–80.
5. Дикарев В.А., Султанов Р.В. Обеспечение качества применения компьютерных систем тренажа. Балашов: Николаев, 2002. 89 с.
6. Пономаренко В.А. Психология жизни и труда летчика. М.: Воениздат, 1992. 58 с.
7. Губсков Ю.А., Прокофьев С.Н. Системная модель информационной конфликтно-устойчивой автоматизированной системы освоения эрготехнических комплексов // Информатика: проблемы, методология, технологии: матер. 12 Междунар. науч.-методич. конф. Воронеж: Изд-во ВГУ, 2012. Т. 1. С. 109–112.
8. Дудоров А.Д. Методика оценки тренажеров лиц группы руководства полетами в интересах повышения качества их подготовки // Современные проблемы и перспективные направления развития авиационных комплексов и систем военного назначения, формы и способы их боевого применения: сб. стат. Всерос. науч.-практич. конф. Воронеж: Изд-во ВАИУ, 2011. С. 92–98.
9. Абу-Абед Ф.Н., Помазуев О.Н., Платонов А.Ю., Мионов А.М., Рюмшин А.Р. Методика определения показателей качества обнаружения радиолокационных станций // Программные продукты и системы. 2014. № 2. С. 118–124.
10. Назаров Т.И. Анализ системы организации тренажной подготовки операторов радиоэлектронных объектов управления воздушным движением // Военная радиоэлектроника: опыт использования и проблемы, подготовка специалистов: тр. XXIV Межвуз. науч.-технич. конф. СПб: Изд-во ВУНЦ ВМФ, 2013. С. 334–338.

UNIFIED DESCRIPTION OF INFORMATION RADIOELECTRONIC SYSTEM FUNCTIONING TO ESTIMATE TRAINING AIDS SOFTWARE

F.N. Abu-Abed¹, Ph.D. (Engineering), Associate Professor, aafares@mail.ru
R.V. Dopira², Ph.D. (Engineering), Professor, Head of the Department, rvdopira@yandex.ru
I.N. Ishchuk³, Ph.D., associate Professor, Head of Department, Boerby@rambler.ru
E.Yu. Brezhnev³, Ph.D. (Engineering), Doctoral Student, dimanbreg@mail.ru
A.N. Potapov³, Ph.D. (Engineering), Associate Professor, Deputy Head of Chair, potapov_il@mail.ru

¹ Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation

² JSC "Research and Production Association Russian basic information technologies" ("NPO RusBITeh"), Kalinina Ave. 17, Tver, 170001, Russian Federation

³ Military Scholastic-Scientific Centre of the Air Forces "Zhukovsky and Gagarin Air Forces Academy", Starykh Bolshevikov St. 54a, Voronezh, 394064, Russian Federation

Abstract. Nowadays, training aids software evaluation is performed according to the coincidence degree of simulated processes with the real one. However, the lack of universality in the description of the mechanisms for operating ergonomic information radio electronic systems (RES), for which training complexes are developed, does not ensure development of tools to form their application strategies taking into account changes in operating conditions that are invariant to RES. As a result, a unified description (representation) of information radio electronic system functioning is relevant. On this basis it is proposed to create software for their training aids.

Considering this, the goal of the work is a unified description of information radio electronic systems functioning with further implementation of their training aids software evaluation. A unified description of information radio electronic systems functioning is based on a unified approach to RES representation as structural and logical schemes. Their distinguishing feature is that they take into account radio frequency (RF) interactions with environmental elements expressed through RF actions and RF influences that form cause-effect relationship of RES operation RF-conditions. The possibilities of implementing RES of the functions of selecting, distributing and redistributing resources depend on the mechanisms of controllability in it. To construct structural and logical schemes of ERT operation that take into account operator activity, the paper considers the mechanisms of their controllability.

This allows us to develop a scientific and methodological instrument to evaluate software of training facilities for air traffic control operators (ATC) based on the examination results. Moreover, when presenting them as complex hierarchical structure systems and in the information field of ATC operator workplaces using Fisher- χ^2 criterion and the generalized distribution law.

Keywords: ergo technical, radio-electronic system, structural and logical, management, model, automated.

References

1. Potapov A.N. *Avtomatizatsiya trenazhnoy podgotovki operatorov radioelektronnykh obektov upravleniya vozdushnym dvizheniem* [Automation Training of Operators of Radio-Electronic Air Traffic Control Facilities]. Voronezh, VAIU Publ., 2010, 136 p.
2. Sysoev E.S. The features of functioning of ergatic electronic equipment for air traffic control. *Nauchnye chteniya imeni A.S. Popova: sb. stat. region. nauch.-praktich. konf.* [Proc. Regional Science and Practice Conf. "Scientific readings n.a. A.S. Popov]. 2012, Voronezh, VUNTS VVS "VVA" Publ., 2012, pp. 38–42 (in Russ.).
3. Lebedev V.V., Romanov A.V. A set-theoretical approach to the functioning description of ergotechnical electronic equipment. *Problemy effektivnosti i bezopasnosti funktsionirovaniya slozhnykh tekhnicheskikh i informatsionnykh sistem: sb. tr. 32 Vseros. nauch.-tekhnich. konf.* [Proc. 32th All-Russian Science and Practice Conf. "Efficiency and Safety Problems of Difficult Technical and Information Systems Functioning"]. 2013, Serpukhov, VA RVS Publ., 2013, pp. 118–122 (in Russ.).
4. Sysoev V.V., Krutskikh P.P., Svintsov A.A. A mathematical model of an information conflict. *Radiotekhnika* [Radioengineering]. 1999, no. 3, pp. 77–80 (in Russ.).
5. Dikarev V.A., Sultanov R.V. *Obespechenie kachestva primeneniya kompyuternykh sistem trenazha* [Quality Assurance in Computer Training System Application]. Balashov, Nikolaev Publ., 2002, 89 p.
6. Ponomarenko V.A. *Psikhologiya zhizni i truda letchika* [Psychology of Pilot's Life and Work]. Moscow: Military Publ., 1992, 58 p.
7. Gubskov Yu.A., Prokofev S.N. A system model of an information conflicts-sustainable automated system for acquisition of ergotechnical complexes. *Informatika: problemy, metodologiya, tekhnologii: mater. 12 Mezhdunar. nauch.-metodich. konf.* [Proc. 12th Int. Science and Methodical Conf. "Computer Science: Problems, Methodology, Technology"]. Voronezh, VGU Publ., 2012, vol. 1, pp. 109–112 (in Russ.).
8. Dudorov A.D. Methodology for the assessment of equipment entities of a flight management group to improve the quality of their training. *Sovremennye problemy i perspektivnye napravleniya razvitiya aviatsionnykh kompleksov i sistem voennogo naznacheniya, formy i sposoby ikh boevogo primeneniya: sb. stat. Vseros. nauch.-praktich. konf.* [Proc. All-Russian Science and Practice Conf. "Modern Problems and Potential Development Directions of Aircraft Systems and Military Systems, Forms and Methods of Their Combat Use"]. Voronezh, VAIU Publ., 2011, pp. 92–98 (in Russ.).
9. Abu-Abed F.N., Pomazuev O.N., Platonov A.Yu., Mironov A.M., Ryumshin A.R. The method of quality factors estimation for radar detection. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 2, pp. 118–124 (in Russ.).
10. Nazarov T.I. The analysis of the training organization system for operators radio electronic objects of air traffic control. *Proc. 24th Interuniversity Science and Tech. Conf. "Military Radio Electronics: Experience and Problems, Training of Specialists"*. 2013, St. Petersburg, VUNC VMF Publ., 2013, pp. 334–338 (in Russ.).

УДК 681.518
DOI: 10.15827/0236-235X.118.307-313

Дата подачи статьи: 26.12.16
2017. Т. 30. № 2. С. 307–313

ПРОГРАММНО-АЛГОРИТМИЧЕСКИЙ КОМПЛЕКС ЗАЩИТЫ И УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

В.Г. Матвейкин, д.т.н., профессор;

Б.С. Дмитриевский, д.т.н., доцент;

В.И. Медников, соискатель;

*С.Г. Семержинский, аспирант, ipu@ahp.tstu.ru
(Тамбовский государственный технический университет,
ул. Советская, 106, г. Тамбов, 392000, Россия)*

В статье описана математическая модель предприятия во взаимодействиях с внешней средой, осуществляемых за счет обмена ресурсами.

Через воздействия на ресурсы происходит как текущее управление предприятием менеджментом, так и реальное несанкционированное влияние на его деятельность со стороны субъекта или внешней среды. Несанкционированные воздействия выявляются при сравнении фактических и расчетных значений показателя деятельности на выбранном уровне исполнения функций. Если такое воздействие определено, система противодействия выполняет свою профильную функцию с целью достижения значения защищенности.

Показатель успешности защиты принят в форме минимальных требований к эффективности на основе статистик или экспертным путем. Найден технический параметр безопасности предприятия, который количественно характеризует неэффективность защиты.

Построена структура системы защиты и управления предприятием, которая выявляет и локализует несанкционированные влияния на его внутренние процессы и процессы взаимодействия с внешней средой без участия человека. Одновременно органично совмещает управление и защиту предприятия, существенно отличается от известных систем. Реализованная как программно-алгоритмический комплекс, система защиты и управления позволяет описывать все производственные процессы.

Предложены средства повышения достоверности и получения измерительной информации о состоянии ресурсов, контролирующие и идентифицирующие внешние влияния на каждый ресурс без участия человека. Разработаны средства получения количественной информации о состоянии ресурсов и найдены условия технической реализации предложенной системы защиты и управления предприятием.

Предложенные алгоритмы безопасного управления используют систему защиты ресурсов, которая надежно обеспечивает достоверность информации на всех уровнях управления. За счет использования представленной системы защиты и управления повышается эффективность предприятия во взаимодействиях с внешней средой.

Ключевые слова: модель предприятия, структура защиты, параметры защиты, риск, производственные ресурсы, субъект влияния, несанкционированное воздействие, показатель успешности защиты, уравнение безопасности ресурсов, угроза.

Вопросы управления предприятиями рассматривались во многих работах, например в [1–8]. Наибольшую актуальность они приобретают с точки зрения безопасности.

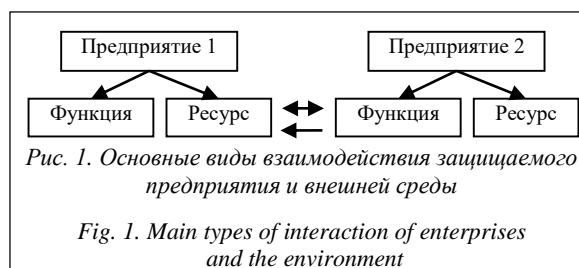
Законность защиты деятельности предприятий продиктована исполнением ими фундаментальной функции $f(s)$ в системе государства, поскольку целями создания любого из предприятий – производственного, оборонного, бюджетного и т.д. – являются удовлетворение потребности внешней среды в их продукции, в пополнении бюджетов разных уровней, в обеспечении благосостояния работающих на них людей.

Состояние предприятия в числе прочих критериев характеризуют его безопасное взаимодействие с внешней средой, влияние и рынок (рис. 1).

Безопасность предприятия закладывается на стадии его проектирования или приобретается во взаимодействиях в результате выполнения им функции защиты. Защита предприятия – его неотъемлемая как функция, так и ресурс; в свою очередь, функция защиты имеет две составляющие: обнаружение негативных влияний на ресурсы (функция

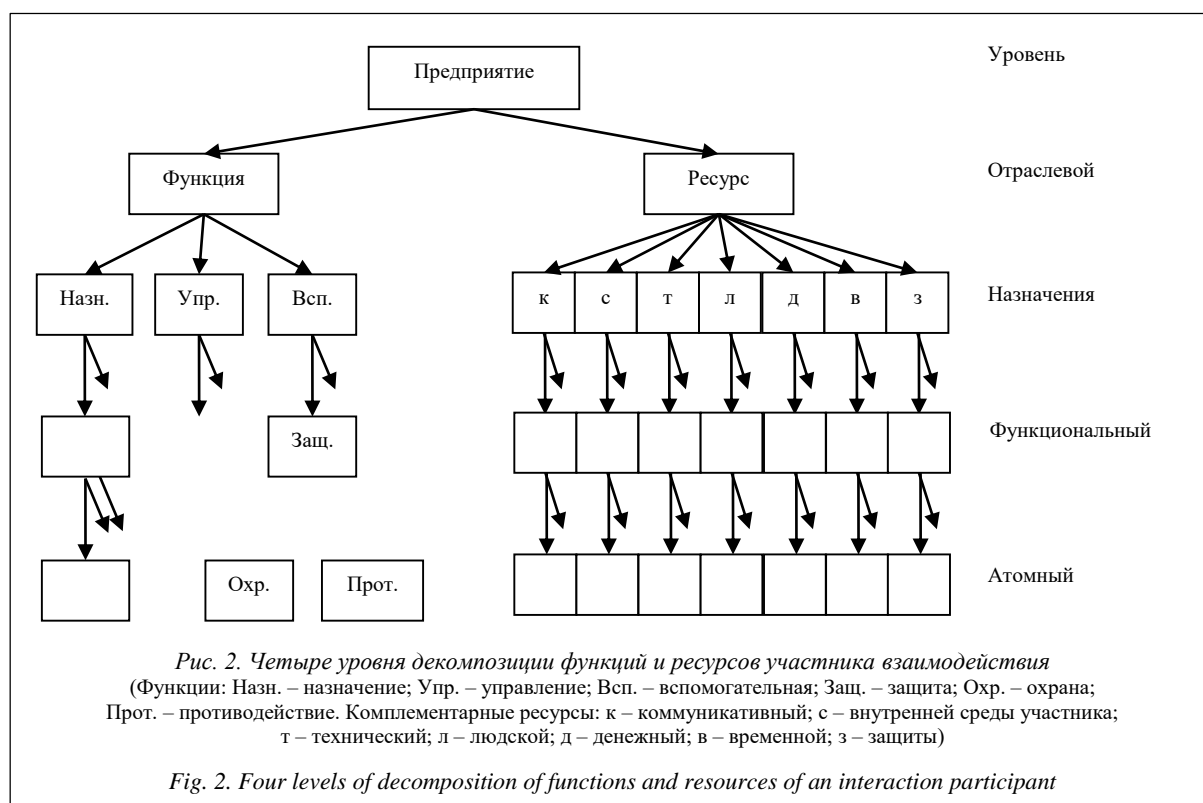
охраны ресурсов) и ликвидация негативных влияний (функция противодействия негативным влияниям).

Внешняя среда любого предприятия – это совокупность предприятий, потребителей, субъектов влияния, производителей со своими функциями назначения (на рисунке 1 внешняя среда – Предприятие 2).



Описание и оценка разработанного подхода

Для определения параметров системы защиты предприятие структурировано путем декомпози-



ции его функций и комплементарных ресурсов [9]. Возможность декомпозиции предприятия (рис. 2) в виде иерархического набора независимых составляющих функций и ресурсов уменьшающейся сложности в одном из основных видов взаимодействия – рынке – проистекает из отсутствия смешанной связности этих составляющих на каждом уровне декомпозиции (гетерогенность). Независимость составляющих следует из самостоятельности содержания каждой из них, специализации процессов, происходящих в них, и обособленности их структуры. В процессе проводимого исследования было установлено, что для другого основного вида взаимодействия – влияния – этой независимости составляющих может и не быть.

Исходя из этого, термин «безопасность предприятия» количественно характеризует пребывание значений выбранного показателя его безопасности (см. табл. 1) в интервале, установленном расчетным, статистическим или экспертным путем.

Исполнение предприятием своей функции на рынке осуществляется путем взаимодействия с внешней средой за счет обмена ресурсами [6]:

$$f(s) = \frac{p_0 + p_1 s^1 + p_2 s^2 + \dots + p_n s^n}{[r_0] + [r_1] s^1 + [r_2] s^2 + \dots + [r_n] s^n} + \phi(s),$$

где $f(s)$ – производственная функция; p_i – факт реализации основного показателя деятельности в i -й момент времени; n – размерность учетного интервала времени $n\Delta t$; Δt – элементарный интервал времени; $[r_i]$ – факт использования комплементарного ресурса для создания p_i ; $[r_i]$ представлены прямоугольной матрицей $[7 \times n]$, где 7 – число строк мат-

рицы, равному количеству ресурсных составляющих предприятия как неотъемлемых, так и привлекаемых извне (рис. 2).

Пример: запись денежной ресурсной составляющей r_{ide} имеет вид матрицы – строки $[r_{ide}]$, $i \in [0, n]$; $s = \text{Re} + j\text{Im}$; Re – абсцисса абсолютной сходимости; Re и Im – аргументы (в данном случае был использован ортонормированный базис $\exp(jn\Delta t/T)$ преобразования Лапласа); $\phi(s)$ – преобразование Лапласа начальных условий деятельности предприятия.

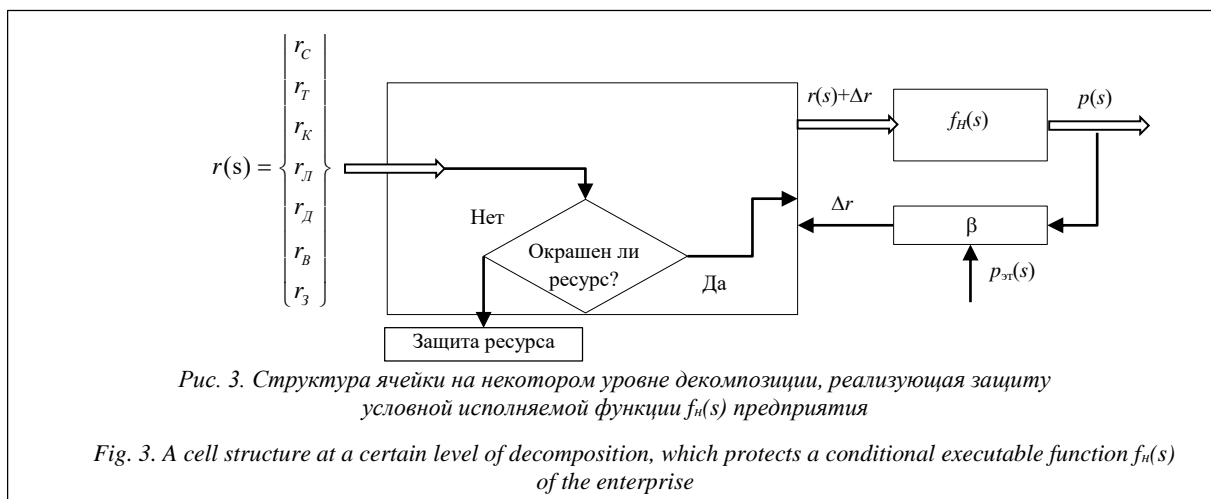
Таблица 1

Соответствие защищаемого показателя предприятия и вида безопасности

Table 1

The correspondence between a protected enterprise index and the type of security

Защищаемый показатель предприятия	Вид безопасности
Основной производственный показатель	Экономическая
Информационный ресурс	Информационная
Периметр территории; доступ персонала	Физическая
Комплексный показатель	Комплексная
Огнестойкость ресурсов	Пожарная
Внутренние производственные процессы	Внутренняя
Производственные вредности	Экологическая
Сейсмическая устойчивость сооружений	Сейсмическая

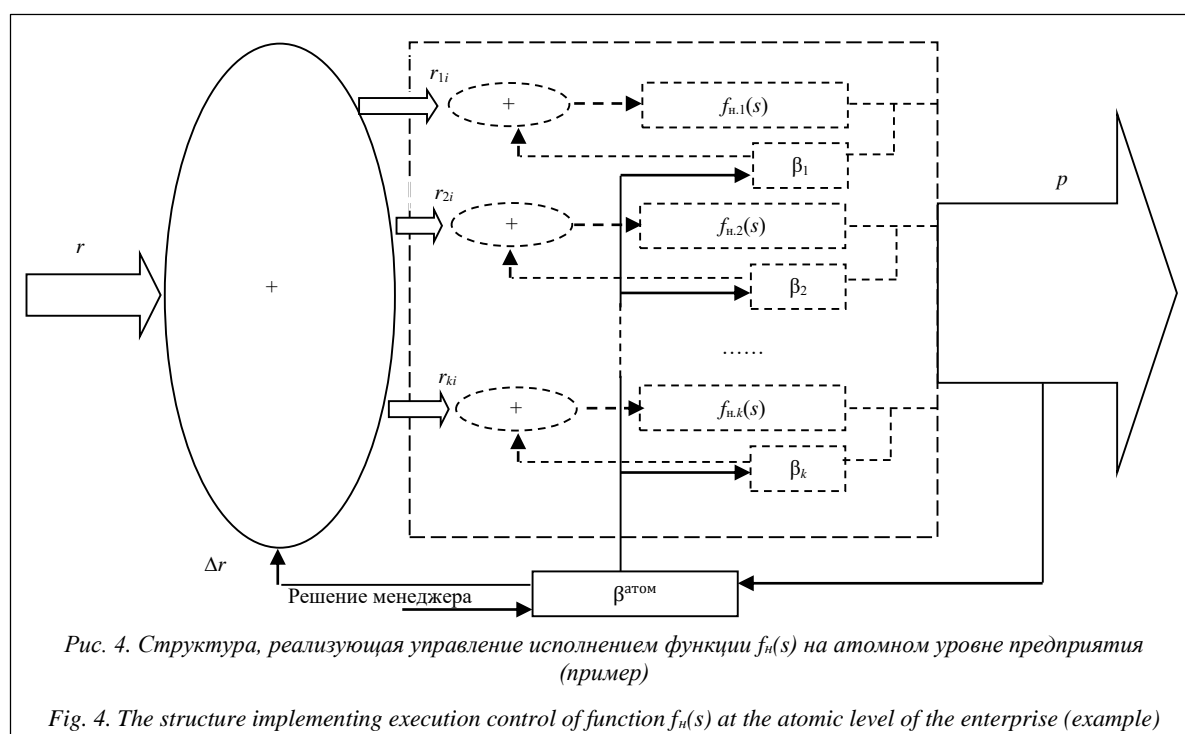


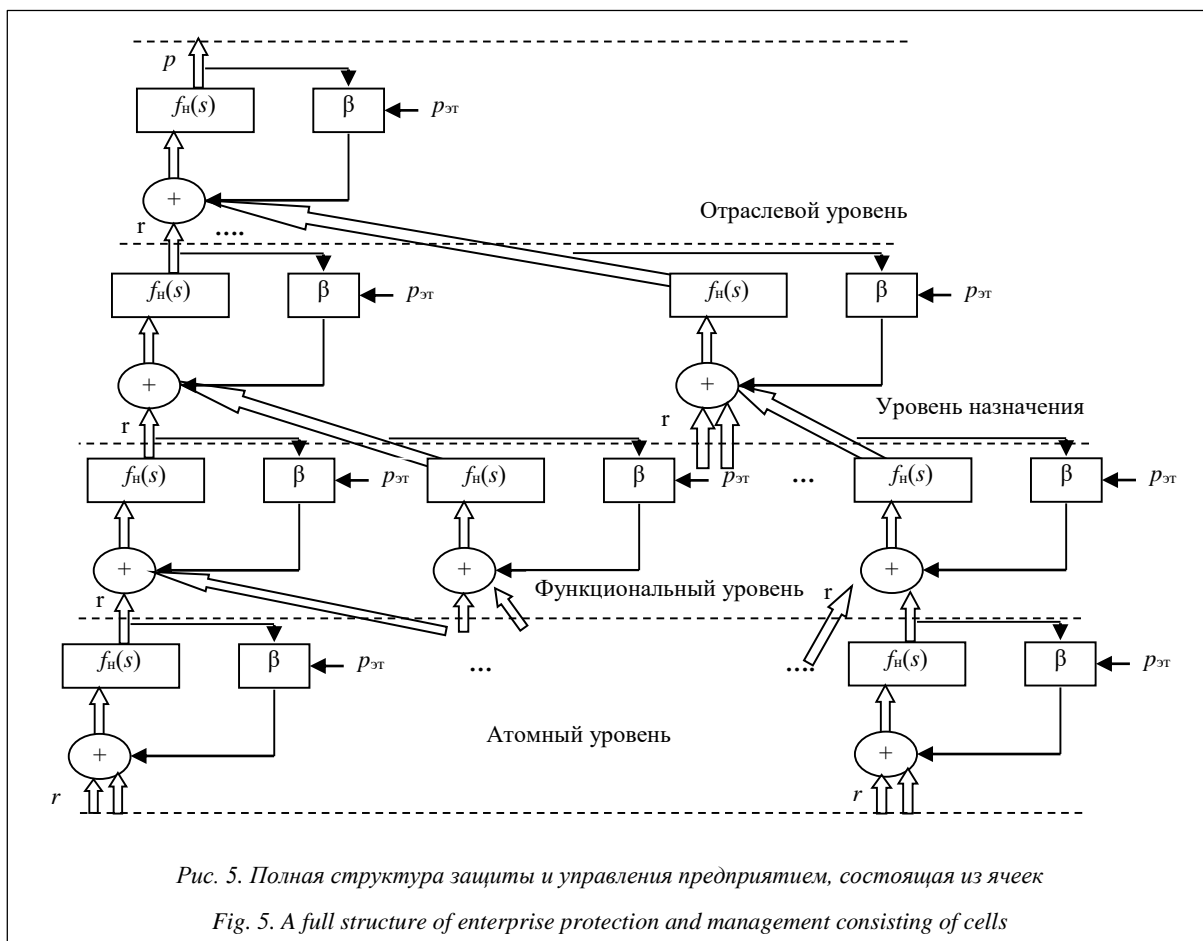
Структура ячейки, реализующей защищенное исполнение каждой функции предприятия на любом уровне его декомпозиции, показана на рисунке 3. На этом и последующих рисунках приняты обозначения: $r(s)$ – комплементарный ресурс; $f_n(s)$ – исполняемая производственная функция, например назначения; $p(s)$ – фактическое значение основного показателя на выбранном уровне; Δr – ресурсная добавка, вырабатываемая в результате управления; $p_{эт}(s)$ – расчетное (эталонное) значение основного показателя на выбранном уровне; β – решение менеджмента о необходимости ресурсной добавки; «Окрашен ли ресурс?» – оценка, вырабатываемая рубежом охраны ресурса по совокупности индикаторов, на выявление санкционированности управляющего воздействия на комплементарный ресурс. Структура, изображенная на

рисунке 4, состоящая из указанных ячеек, представляет общий случай реализации функции управления предприятием на атомном уровне его декомпозиции. Полная структура предприятия, реализующая многоуровневую среду создания основного показателя деятельности $p(s)$ [10], изображена на рисунке 5.

На рисунке 6 показано взаимодействие одной из ячеек (Яч 1-2) полной структуры защиты и управления предприятием (см. рис. 5) с контроллером РУ (например, на базе PIC-контроллера) атомного уровня исполнения функции $f_n(s)$ (см. рис. 4) с целью повышения достоверности информации о первичном производственном процессе защищаемого предприятия.

Достоверное распознавание каждой ячейкой влияний на ресурсы приблизило программную об-



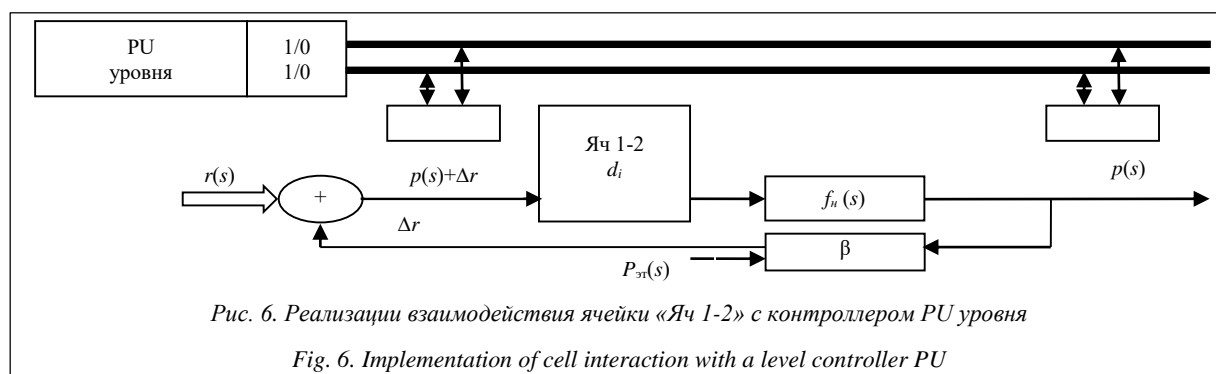


работку информации к первичным процессам и позволило использовать 1 бит для отражения состояния каждой из них в информационной части квитанции, выставляемой контроллеру РУ. Следствием такого приближения программной обработки является прозрачность исполнения функций на всех уровнях управления. Алгоритм квитирования запросов уровневого контроллера представлен по ссылке http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/5.jpg/.

Существенные отличия функциональных возможностей предложенной системы безопасного управления: 1) каждая ячейка любого уровня имеет обособленную структуру, цель и процесс; 2) ячейка общается с контроллером уровня по протоколу CDDI для повышения надежности информации о

внутренних процессах по разработанной программе квитирования запросов [11]; 3) взаимодействие ячеек соседних уровней происходит посредством защищенного коммуникативного ресурса, что обеспечивает искомое повышение надежности управления предприятием в целом за счет уменьшения вероятности ее катастрофического отказа; 4) система выполняет свои функции, начиная строго с первичных производственных процессов.

Информационная поддержка при принятии управленческих решений реализована путем распределения информации о внутренних процессах, о состоянии его ресурсов на всех уровнях между управленцами, а также организацией информационного обмена между хостом защищаемого предприятия и хостами предприятий-поставщиков сети.



Обсуждение проблемы

Не вызывает сомнений утверждение: как реальное текущее управление предприятием со стороны менеджмента, так и реальные несанкционированные влияния на его деятельность со стороны внешней среды или субъекта влияния производятся путем воздействия на ресурсы как средства осуществления его функций. Вследствие таких изменений ресурсов исполняемые предприятием функции также изменяются: санкционированно при управлении и несанкционированно при влияниях. Поэтому авторы сфокусировали интерес на определении границ достаточности защиты предприятия с точки зрения как его внутренних процессов, так и процессов во внешней среде.

Практика деятельности предприятий показывает, что защищенность производственных ресурсов (как собственных, так и привлекаемых извне) недостаточна. Подсистема охраны ресурсов (одна из составных частей системы защиты) выявляет несанкционированное воздействие v_n (v – variety, n – негативное) на них по результату сравнения фактических и расчетных значений показателя $pi(s)$ деятельности на выбранном уровне исполнения функций. Если v_n выявлено, подсистема противодействия (другая составная часть системы защиты) выполняет свою профильную функцию, оказывая ему противодействие a_n (a – anti-variety) с целью достижения некоторого защищенного значения $p_3(s)$. Процедуру противодействия и его количественный результат на этом уровне формализуют уравнение защищенности во влияниях $p_3(s) = a_n v_n r(s) f(s)$, которое совпадает с основным уравнением безопасности на рынке, поэтому является его версией, его уточнением.

Практика показала: уже сейчас несложным для специалиста по защите предприятия является формирование аналитических записей влияний на каждую составляющую ресурса на каждом уровне исполнения функций (рис. 2) в виде $v_n a_n$. Например, для атомного уровня ресурса среды такое влияние v_n с есть появление вредностей в воздушной среде на рабочем месте в количестве, превышающем предельно допустимую концентрацию. Адекватным противодействием a_n с такому превышению является увеличение интенсивности воздухообмена в этой среде путем вентиляции. Аналогичные процедуры обеспечения безопасности выполняются для остальных ресурсных составляющих на остальных уровнях предприятия.

Показателем успешности защиты (охраны ресурсов и противодействия негативным влияниям на них) ресурсной составляющей x авторами принята эффективность: $E_{3,x} = E_{охр,x} E_{n,x}$, количественное наполнение которого не унифицировано для всех уровней предприятия и не установлено на межотраслевом уровне по причине участия в его формировании человеческого фактора. Поэтому в настоя-

щее время он принят в форме минимальных требований к эффективности на основе статистик или экспертным путем.

Таким образом, технический параметр безопасности предприятия можно записать в виде $1 - E_{3,x} = 1 - E_{охр,x} E_{n,x}$. Он количественно характеризует неэффективность защиты, или безнаказанность несанкционированного влияния на ресурсную составляющую x . На его основе можно записать уравнение безопасности ресурсов $r_{безоп}(s)$: $r_{безоп}(s) = E_{П,с} E_{охр,с} r_c(s) + E_{П,к} E_{охр,к} r_k(s) + E_{П,т} E_{охр,т} r_t(s) + E_{П,л} E_{охр,л} r_l(s) + E_{П,д} E_{охр,д} r_d(s) + E_{П,в} E_{охр,в} r_v(s) + E_{П,з} E_{охр,з} r_z(s)$, в котором мультипликаты $E_{П,с} E_{охр,с}$ означают, что ни охрана, ни противодействие не компенсируют друг друга при защите составляющих, то есть не являются взаимодополняемыми.

Полученное выражение для $r_{безоп}(s)$ дает защищенные (безопасные) значения основного показателя деятельности $p_{безоп}(s) = f(s) r_{безоп}(s)$ на выбранном уровне исполнения функции (структуры) предприятия во взаимодействиях.

Найденное уравнение безопасности позволило рассчитывать риск предприятия во взаимодействиях с внешней средой или, другими словами, относительный размер потерь ресурсов из-за несовершенства исполнения функции охраны и исполнения функции противодействия. Величина риска количественно определена, очевидно, разностью между расчетным $p(s)$ (эталонным) и безопасным $p_{безоп}(s)$ значениями основного показателя деятельности: $risk p(s) = p(s) - p_{безоп}(s) = p(s) - E_{охр} E_{П,x} p(s) = (1 - E_{охр} E_{П,x}) p(s)$, где $(1 - E_{охр} E_{П,x})$ – искомый риск предприятия во взаимодействиях с внешней средой.

Несовершенство современных технических средств защиты вынуждает менеджмент предприятий использовать человека в контуре защиты ресурсов. Попутно отметим: в теории экономических игр априори принятая 100 %-ная эффективность каждой из подсистем защиты ($E_{охр} = 1$, $E_3 = 1$ или нуль-эффект влияний) заставила отказаться от использования этой теории для решения задачи исследования.

Найденное выражение для риска приводит к количественному определению величины угрозы $Y(s)$ или $th(s)$ (threat – угроза) защищаемому ресурсу предприятия через эффективности $E_{св}$ и E_3 : угроза $Y(s)$ – это риск $(1 - E_{П} E_{охр})$, превысивший порог $(1 - E_{П} E_{охр})_{пор}$, установленный экспертным или иным путем. В рассматриваемом влиянии субъекта влияния на предприятие численное значение такого порога дает эффективность $E_{св}$ исполнения субъектом влияния своей функции назначения. Для такого влияния справедливо неравенство $Y(s) > (1 - E_{П} E_{охр})_{пор}$ при $(1 - E_{П} E_{охр})_{пор} = E_{св}$, где $E_{св}$ – эффективность субъекта влияния. Такая найденная в исследовании формула угрозы обобщает различные версии этого термина.

Примеры численных значений риска и угроз для разных видов взаимодействия в таблице 2 показывают: скрытый ущерб влияний (риск) не превращается в явный (угроза) до того момента, когда $r_3 > r_{св}$.

Таблица 2

Численные значения риска и угроз

Table 2

Numerical values of a risk and threats

Взаимодействие	Порог	Риск	Угроза
Землетрясение	3 балла	Меньше 3 баллов	Больше (или равно) 3 баллов
Заболеваемость персонала	30 %	Меньше 30 %	Больше (или равно) 30 %
Эксплуатационная надежность оборудования	Предельное состояние	Эксплуатация до достижения предельного состояния	Эксплуатация после достижения предельного состояния
Заем кредита в банке	10 % численности заемщиков	Меньше 10 % численности заемщиков	Больше (или равно) 10 % численности заемщиков

Заключение

Рассмотренные примеры показывают, что, во-первых, безопасность предприятия достигается защитой каждой отдельной ресурсной составляющей на всех уровнях его структуры даже при нали-

чии во внешней среде негативных влияний, эффективность которых меньше эффективности защиты, а во-вторых, найденное определение угрозы можно распространить на влияние предприятия на себя.

Литература

1. Попов Н.С., Лузгачева Н.В., Чан Минь Тьинь. К методике конструирования экспертной системы оценки промышленной безопасности // Вестн. ТГТУ. 2013. Т. 19. № 1. С. 43–51.
2. Битюков В.К., Емельянов А.Е. Качественный анализ функционирования сетевой системы управления с конкурирующим методом доступа // Вестн. ТГТУ. 2012. Т. 18. № 1. С. 38–46.
3. Битюков В.К., Емельянов А.Е. Обобщенная математическая модель сетевой системы управления с конкурирующим методом доступа // Вестн. ТГТУ. 2012. Т. 18. № 2. С. 319–326.
4. Дякин В.Н. Динамическая модель управления развитием промышленного предприятия // Вестн. ТГТУ. 2013. Т. 19. № 2. С. 304–308.
5. Серов А.Ю., Сморгонский А.В. Действующая компьютерная модель производственного предприятия // Экономика и математические методы. 2009. Т. 45. № 3. С. 40–47.
6. Еремеев А.П. Прототип интеллектуальной системы поддержки принятия решений для управления энергообъектом // Программные продукты и системы. 2002. № 3. С. 38–42.
7. Колесников А.А. Синергетические методы управления сложными системами: теория системного синтеза. М.: Либроком, 2012. 240 с.
8. Литвак Б.Г. Разработка управленческого решения. М.: Дело, 2004. 392 с.
9. Медников В.И., Орехов С.А. Эффективное управление предприятием // Экономика и управление. 2007. № 2. С. 101–106.
10. Mednikov B.V., Mednikov V.I., Mednikov S.V. Commodity market math models // Экономика, статистика и информатика. Вестник УМО. 2015. № 2. С. 194–198.
11. Медников В.И. Квотирование запросов уровневого контроллера. Свид. об офиц. регистр. программы для ЭВМ № 2016619055.2016.

Software & Systems

DOI: 10.15827/0236-235X.118.307-313

Received 26.12.16

2017, vol. 30, no. 2, pp. 307–313

A SOFTWARE AND ALGORITHM COMPLEX OF ENTERPRISE PROTECTION AND MANAGEMENTV.G. Matveykin¹, Dr.Sc. (Engineering), ProfessorB.S. Dmitrievsky¹, Dr.Sc. (Engineering), Associate ProfessorV.I. Mednikov¹, CandidateS.G. Semerzhinsky¹, Postgraduate Student, ipu@ahp.tstu.ru¹ Tambov State Technical University, Sovetskaya St. 106, Tambov, 392000, Russian Federation

Abstract. The paper presents an enterprise mathematical model interacting with external environment by resource exchange.

Current enterprise management, as well as real unauthorized influence on its activities by external environment or a subject is a result of the impact on resources. Unauthorized actions are detected by comparison of actual and calculated values of performance indicators for a selected level of functions execution. If such action is detected, a countermeasure system becomes active to reach the protected value.

The protection success rate is accepted as minimum performance requirements based on statistics and expert way. As a result, the authors have found a technical parameter of enterprise security that quantitatively describes protection inefficiency.

The built structure of enterprise protection and management system detects and locates unauthorized effect on its internal processes and the processes of interaction with the environment without human intervention. At the same time it organically combines enterprise management and protection and is significantly different from known systems. The protection and control system, which is implemented as software and algorithm complex, makes it possible to describe all production processes.

The paper proposes the means of improving reliability of measurement and obtaining information about resources state. They monitor and identify the external influences on each resource without human intervention. The paper describes the developed means of obtaining quantitative information on resource state and the terms of technical feasibility of the proposed enterprise protection and management system.

The proposed secure management algorithms use a resource protection system that provides reliable information at all management levels. The presented the protection and control system raise the enterprise efficiency in the interactions with the environment.

Keywords: enterprise model, protection structure, protection parameters, risk, productive resources, influence subject, unauthorized influence, protection success rate, resource security equation, threat.

References

1. Popov N.S., Luzgacheva N.V., Chan M.T. To methods of designing expert evaluation system of industrial safety. *Vestn. TGTU* [Trans. of the TSTU]. 2013, vol. 19, no. 1, pp. 43–51 (in Russ.).
2. Bityukov V.K., Emelyanov A.E. Qualitative Analysis of Network Control System with Competing Access Mode. *Vestn. TGTU* [Trans. of the TSTU]. 2012, vol. 18, no. 1, pp. 38–46 (in Russ.).
3. Bityukov V.K., Emelyanov A.E. Generalized Mathematical Model of Network Control System with Competing Access Mode. *Vestn. TGTU* [Trans. of the TSTU]. 2012, vol. 18, no. 2, pp. 319–326 (in Russ.).
4. Dyakin V.N. Dynamic Model of Management of Industrial Enterprise Development. *Vestn. TGTU* [Trans. of the TSTU]. 2013, vol. 19, no. 2, pp. 304–308 (in Russ.).
5. Serov A.Yu., Smorgonsky A.V. Acting computer model of the manufacturing enterprise. *Ekonomika i matematicheskie metody* [Economics and Mathematical Methods]. 2009, vol. 45, no. 3, pp. 40–47 (in Russ.).
6. Eremeev A.P. The prototype of intellectual decision support system for the management of power facilities. *Programmnye produkty i sistemy* [Software & Systems]. 2002, no. 3, pp. 38–42 (in Russ.).
7. Kolesnikov A.A. *Sinergeticheskie metody upravleniya slozhnymi sistemami. Teoriya sistemnogo sinteza* [Synergetic Control Methods of Complex Systems. System Synthesis Theory]. Moscow, Librokom Publ., 2012, 240 p.
8. Litvak B.G. *Razrabotka upravlencheskogo resheniya* [Development of a Management Solution]. Moscow, Delo Publ., 2004, 392 p.
9. Mednikov V.I., Orekhov S.A. Effective enterprise management. *Ekonomika i upravlenie* [Economics and Management]. 2007, no. 2, pp. 101–106 (in Russ.).
10. Mednikov B.V., Mednikov V.I., Mednikov S.V. Commodity market math models. *Ekonomika, statistika i informatika. Vestnik UMO* [Economics, Statistics and Informatics. Bulletin of UMO]. 2015, no. 2, pp. 194–198 (in Russ.).
11. Mednikov V.I. *Kvitirovanie zaprosov urovnevnogo kontrollera* [Level Controller Challenge Handshake]. Official Registration Certificate of the Computer Program no. 2016612717. 2016.

Примеры библиографического описания статьи

1. Матвейкин В.Г., Дмитриевский Б.С., Медников В.И., Семержинский С.Г. Программно-алгоритмический комплекс защиты и управления предприятием // Программные продукты и системы. 2017. Т. 30. № 2. С. 307–313; DOI: 10.15827/0236-235X.118.307-313.
2. Matveykin V.G., Dmitrievsky B.S., Mednikov V.I., Semerzhinsky S.G. A software and algorithm complex of enterprise protection and management. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 307–313 (in Russ.); DOI: 10.15827/0236-235X.118.307-313.

УДК 004.056

DOI: 10.15827/0236-235X.118.314-319

Дата подачи статьи: 02.10.16

2017. Т. 30. № 2. С. 314–319

МЕТОД ФОРМИРОВАНИЯ МНОЖЕСТВ АЛЬТЕРНАТИВНЫХ ВАРИАНТОВ ПОСТРОЕНИЯ ПОДСИСТЕМ, ВХОДЯЩИХ В СОСТАВ СИСТЕМЫ ЗАЩИТЫ ОТ КОМПЬЮТЕРНЫХ АТАК

Е.Б. Дроботун, к.т.н., докторант, drobotun@xakep.ru;

Е.П. Угловский, начальник лаборатории;

И.Ш. Замалтдинов, к.т.н., старший научный сотрудник

(Военная академия воздушно-космической обороны им. Маршала Советского Союза Г.К. Жукова, ул. Жигарева, 50, г. Тверь, 170100, Россия)

Построение рациональной системы защиты от компьютерных атак для информационно-вычислительной или автоматизированной системы предполагает формирование множества конфигураций системы защиты, состоящей, в свою очередь, из множества отдельных программных и программно-аппаратных компонентов, и дальнейший выбор из сформированного множества рационального варианта построения системы защиты от компьютерных атак по определенным критериям. При формировании данного множества, помимо соответствия системы защиты необходимым функциональным требованиям, следует учитывать как параметры самой защищаемой системы (ее структуру и многоуровневость построения), так и программную и аппаратную совместимость компонентов между собой, а также совместимость компонентов с программно-аппаратной платформой, на базе которой построена защищаемая система.

В статье представлен один из подходов к формированию множества возможных вариантов построения системы защиты от компьютерных атак с учетом ее декомпозиции на три подсистемы: подсистему обнаружения компьютерных атак, подсистему противодействия компьютерным атакам и подсистему устранения последствий применения компьютерных атак.

Ключевые слова: автоматизированная система управления, компьютерная атака, информационная безопасность, проектирование системы защиты, защита от компьютерных атак.

Решение задачи структурно-параметрического синтеза системы защиты информационно-вычислительных и автоматизированных систем различного назначения от компьютерных атак заключается в выборе наилучших вариантов системы защиты в условиях ограничений и предполагает наличие множества альтернативных вариантов ее построения.

С целью уменьшения количества вариантов, упрощения их формирования и проведения расчетов показателей, характеризующих эти варианты, необходимо осуществить декомпозицию задачи выбора рационального варианта построения системы защиты от компьютерных атак. Как показано в [1], декомпозицию целесообразно осуществлять как по последовательности выполнения процедур формирования и выбора вариантов, так и по подсистемам системы защиты информационно-вычислительных и автоматизированных систем от компьютерных атак.

В целом систему защиты от компьютерных атак для информационно-вычислительных и автоматизированных систем можно декомпозировать на три подсистемы: подсистему обнаружения компьютерных атак, подсистему противодействия компьютерным атакам и подсистему устранения последствий применения компьютерных атак. При этом при декомпозиции по последовательности выполнения процедур формирования и выбора вариантов формирование рационального варианта построения системы защиты в целом необходимо начать с формирования множества альтернативных вариан-

тов построения подсистем, входящих в систему защиты от компьютерных атак.

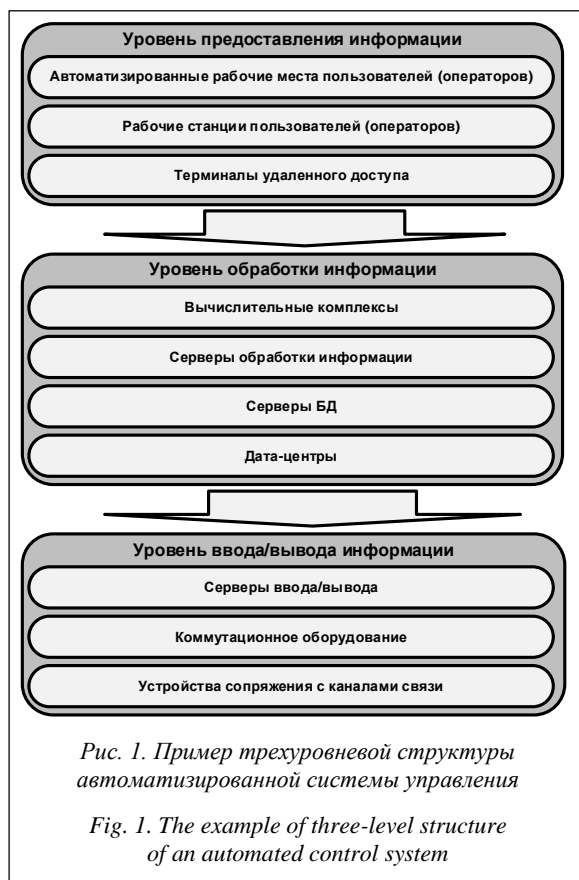
Формирование множества альтернативных вариантов построения подсистем, входящих в систему защиты от компьютерных атак, осуществляется в два этапа:

- формирование полного множества вариантов построения подсистем, входящих в систему защиты от компьютерных атак;
- отбор из сформированного множества вариантов построения подсистем технически реализуемых для конкретной конфигурации защищаемой информационно-вычислительной или автоматизированной системы.

Формирование полного множества вариантов построения подсистем, входящих в систему защиты от компьютерных атак

Полное множество вариантов построения каждой из подсистем, входящих в систему защиты от компьютерных атак, для информационно-вычислительных и автоматизированных систем $V = \{V_{\text{обн}}, V_{\text{пр}}, V_{\text{устр}}\}$ целесообразно формировать путем решения комбинаторной задачи перебора всех возможных компонентов, которые могут входить в состав каждой из подсистем, и задач, возложенных на подсистему, с учетом многоуровневости построения защищаемой информационно-вычислительной или автоматизированной системы [2] (например, уровень представления информации (операторский),

уровни обработки информации и ввода/вывода информации) (рис. 1).



Тогда формирование множеств вариантов построения подсистем, входящих в систему защиты от компьютерных атак, будет включать:

- формирование множества вариантов построения подсистемы для каждого из уровней с учетом задач, возложенных на подсистему для каждого уровня защищаемой информационно-вычислительной или автоматизированной системы;
- формирование множества вариантов построения подсистемы для всех уровней защищаемой информационно-вычислительной или автоматизированной системы в целом из множеств, полученных на предыдущем этапе.

Формирование множества вариантов построения подсистемы, входящей в состав системы защиты от компьютерных атак, для каждого уровня защищаемой системы. Необходимость отдельного формирования множества возможных вариантов построения подсистем системы защиты от компьютерных атак для каждого уровня построения защищаемой системы возникает вследствие, во-первых, разных функциональных требований к каждой из подсистем для каждого уровня построения защищаемой информационно-вычислительной или автоматизированной системы, а во-вторых, построения каждого уровня защищаемой системы на разных программно-аппаратных плат-

формах и, соответственно, с разным составом компонентов средств защиты для каждой из подсистем. Исходя из этого множество всех возможных вариантов V построения подсистем, входящих в состав системы защиты от компьютерных атак, будет выглядеть следующим образом: $V = \{\{V_{обн1}, V_{обн2}, \dots, V_{обнl}\}, \{V_{пр1}, V_{пр2}, \dots, V_{прl}\}, \{V_{устр1}, V_{устр2}, \dots, V_{устрl}\}\}$, где l – количество уровней построения защищаемой системы.

В общем, формирование множества вариантов построения подсистемы для одного уровня построения защищаемой информационно-вычислительной (автоматизированной) системы осуществляется в несколько этапов.

Этап 1. Построение таблицы выполнения требований каждым компонентом средств защиты, входящим в подсистему, для рассматриваемого уровня (табл. 1). В данной таблице отражены функциональные требования ($f_j \in F_k$, $j = \overline{1, J}$, где J – число функциональных требований) для рассматриваемой k -й подсистемы соответствующего уровня к отдельным компонентам ($s_i \in S_k$, $i = \overline{1, I}$, где I – число компонентов средств защиты), входящим в состав рассматриваемой k -й подсистемы соответствующего уровня.

Элементы данной таблицы формируются с учетом следующего условия:

$$v_{ij} = \begin{cases} 1, & \text{если } i\text{-й компонент удовлетворяет} \\ & j\text{-му требованию,} \\ 0, & \text{если } i\text{-й компонент не удовлетворяет} \\ & j\text{-му требованию.} \end{cases}$$

Таблица 1

Анализ выполнения требований каждым компонентом средств защиты для каждого уровня защищаемой системы

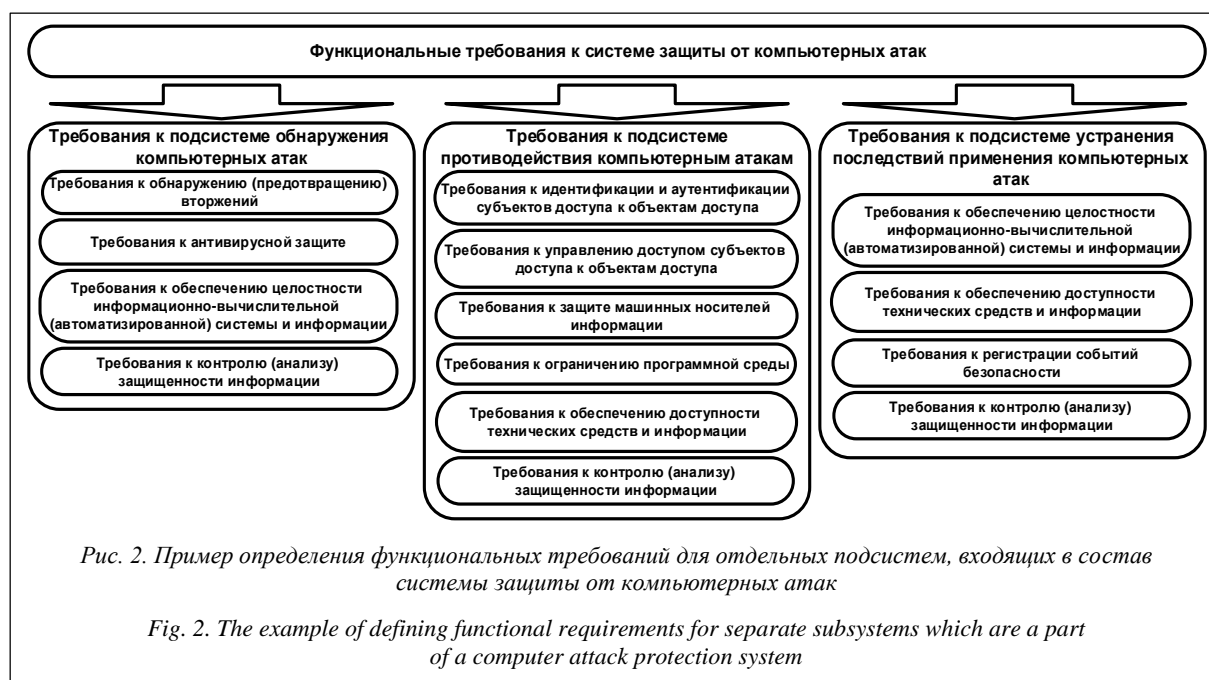
Table 1

The analysis of compliance with requirements by each component of protection means for each level of a protected system

Отдельные компоненты	Функциональные требования					
	f_1	f_2	f_3	f_4	...	f_l
s_1	v_{11}	v_{12}	v_{13}	v_{14}	...	v_{1J}
s_2	v_{21}	v_{22}	v_{23}	v_{24}	...	v_{2J}
s_3	v_{31}	v_{32}	v_{33}	v_{34}	...	v_{3J}
s_4	v_{41}	v_{42}	v_{43}	v_{44}	...	v_{4J}
...
s_l	v_{l1}	v_{l2}	v_{l3}	v_{l4}	...	v_{lJ}

При этом критерии для каждой подсистемы определяются отдельно и, к примеру, согласно [3–8], могут включать в себя функциональные требования

- к идентификации и аутентификации субъектов доступа к объектам доступа;
- к управлению доступом субъектов доступа к объектам доступа;
- к ограничению программной среды;
- к защите машинных носителей информации;



- к регистрации событий безопасности;
- к антивирусной защите информации;
- к обнаружению (предотвращению) вторжений;
- к контролю (анализу) защищенности информации;
- к обеспечению целостности информационно-вычислительной (автоматизированной) системы и информации;
- к обеспечению доступности информации.

Возможный вариант определения требований для отдельных подсистем, входящих в состав системы защиты от компьютерных атак, показан на рисунке 2.

Этап 2. Проверка всей совокупности компонентов на выполнение функциональных требований. Данный этап реализуется следующим образом (рис. 3):

- осуществляется поэлементная дизъюнкция каждой строки из полученной на первом этапе таблицы: $\{v_1, v_2, v_3, \dots, v_J\} = \{v_{11}, v_{12}, v_{13}, \dots, v_{1J}\} \vee \{v_{21}, v_{22}, v_{23}, \dots, v_{2J}\} \vee \dots \vee \{v_{I1}, v_{I2}, v_{I3}, \dots, v_{IJ}\}$;
- выполняется конъюнкция всех элементов получившегося множества $\{v_1, v_2, v_3, \dots, v_J\}$: $v = v_1 \& v_2 \& v_3 \& \dots \& v_J$;
- проверяется условие равенства получившегося результата (v) единице: если данное условие выполняется ($v = 1$), вся совокупность компонентов средств защиты подсистемы удовлетворяет функциональным требованиям; в противном случае ($v \neq 1$) компонентов средств защиты, из которых может быть построена подсистема, недостаточно, и поэтому необходимо расширять весь возможный спектр применяемых компонентов средств защиты

для рассматриваемой подсистемы соответствующего уровня.

Этап 3. Выбор вариантов построения подсистемы, входящей в состав системы защиты от компьютерных атак, для соответствующего уровня построения защищаемой информационно-вычислительной или автоматизированной системы. Суть данного этапа в отборе конфигураций компонентов подсистемы, которые позволят реализовать все функциональные требования к подсистеме.

В таблице 2 показан пример анализа выполнения функциональных требований к одной из подсистем системы защиты.



Таблица 2

Пример анализа выполнения функциональных требований к одной из подсистем системы защиты для отдельного уровня защищаемой системы

Table 2

The example of the analysis of implementing functional requirements to one of protection system subsystems for a separate level of the protected system

Отдельные компоненты, с помощью которых возможна реализация k -й подсистемы S_k	Функциональные требования					
	f_1	f_2	f_3	f_4	f_5	f_6
s_1	1	1	0	1	0	0
s_2	1	0	1	0	1	0
s_3	0	1	0	0	0	1
s_4	1	0	0	0	0	1
s_5	0	1	1	1	0	1

Для данного примера множество всех возможных компонентов, с помощью которых возможна реализация k -й подсистемы, включает в себя пять элементов: $S_k = \{s_1, s_2, s_3, s_4, s_5\}$.

Функциональные требования в данном случае заданы множеством из шести элементов: $F_k = \{f_1, f_2, f_3, f_4, f_5, f_6\}$.

Отбор возможных конфигураций компонентов k -й подсистемы для соответствующего уровня, которые позволяют реализовать все функциональные требования к данной подсистеме, осуществляется следующим образом:

- формируются все возможные варианты комбинаций компонентов, из которых возможно построение k -й подсистемы для рассматриваемого уровня защищаемой подсистемы; при этом общее количество всех возможных вариантов $N=2^m-1$, где m – количество всех компонентов, из которых возможно построение k -й подсистемы;

- проверяется выполнение функциональных требований каждого варианта комбинаций компонентов, из которых возможно построение k -й подсистемы для рассматриваемого уровня защищаемой системы, путем поэлементной дизъюнкции строк, относящихся к компонентам, входящим в анализируемый вариант, далее выполняются поэлементная конъюнкция получившегося множества и проверка выполнения условия равенства полученного результата единице (рис. 4); при этом для каждого элемента строки осуществляется операция конъюнкции с числом t , которое получается путем сдвига вправо на один разряд значения n (операция $\text{shr}(n)$) и логического умножения результата этой операции на 1.

В качестве примера можно рассмотреть проверку на выполнение функциональных требований нескольких вариантов конфигурации применительно к таблице 2.

Пример 1. Конфигурация в составе $\{s_1, s_2, s_3, s_4\}$. Производим поэлементную дизъюнкцию строк, относящихся к компонентам s_1, s_2, s_3 и s_4 : $v = \{1, 1, 0,$

$1, 0, 0\} \vee \{1, 0, 1, 0, 1, 0\} \vee \{0, 1, 0, 0, 0, 1\} \vee \{1, 0, 0, 0, 0, 1\} = \{1, 1, 1, 1, 1, 1\}$.

Осуществляем поэлементную конъюнкцию получившегося множества v и получаем единицу. Таким образом, данная конфигурация позволяет выполнить все функциональные требования, предъявленные к анализируемой подсистеме.

Пример 2. Конфигурация в составе $\{s_1, s_3, s_4, s_5\}$. Производим поэлементную конъюнкцию строк, относящихся к компонентам s_1, s_3, s_4 и s_5 : $v = \{1, 1, 0, 1, 0, 0\} \vee \{0, 1, 0, 0, 0, 1\} \vee \{1, 0, 0, 0, 0, 1\} \vee \{0, 1, 1, 0, 1, 1\} = \{1, 1, 1, 1, 0, 1\}$.

Осуществляем поэлементную конъюнкцию получившегося множества v и получаем ноль. Таким образом, данная конфигурация не позволяет выполнить все функциональные требования, предъявленные к анализируемой подсистеме.

Для случая, приведенного в таблице 2, возможны 11 вариантов построения k -й подсистемы (табл. 3).

Таблица 3

Пример вариантов построения k -й подсистемы для отдельного уровня защищаемой системы

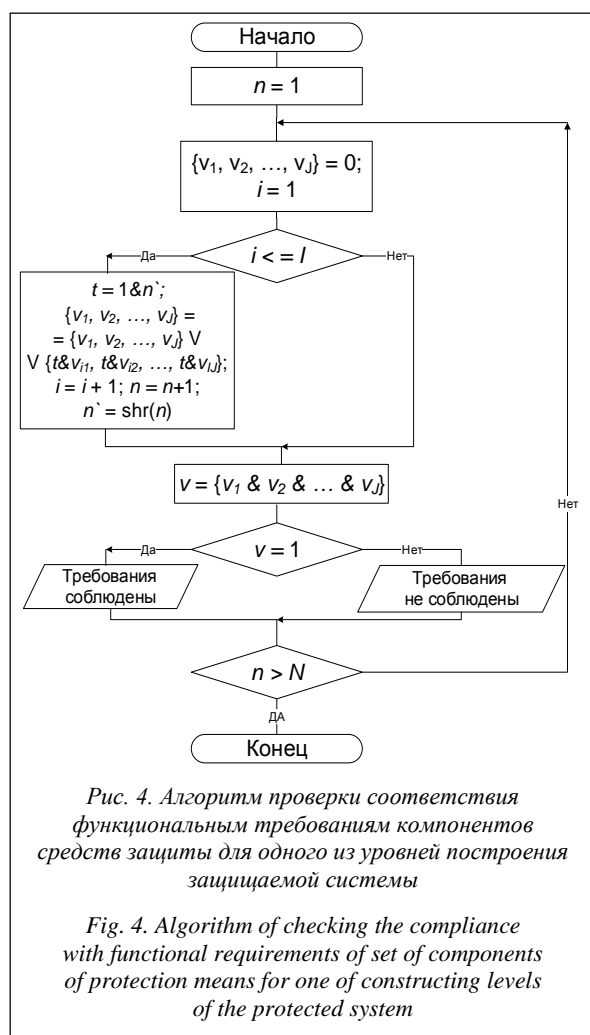
Table 3

The example of k -th subsystem constructing variants for a particular level of the protected system

Вариант построения k -й подсистемы	Компоненты, входящие в вариант построения k -й подсистемы
S_{k1}	$\{s_1, s_2, s_3\}$
S_{k2}	$\{s_1, s_2, s_4\}$
S_{k3}	$\{s_1, s_2, s_3, s_4\}$
S_{k4}	$\{s_2, s_5\}$
S_{k5}	$\{s_1, s_2, s_5\}$
S_{k6}	$\{s_2, s_3, s_5\}$
S_{k7}	$\{s_1, s_2, s_3, s_5\}$
S_{k8}	$\{s_2, s_4, s_5\}$
S_{k9}	$\{s_1, s_2, s_4, s_5\}$
S_{k10}	$\{s_2, s_3, s_4, s_5\}$
S_{k11}	$\{s_1, s_2, s_3, s_4, s_5\}$

Формирование множества вариантов построения подсистем, входящих в систему защиты от компьютерных атак, для всех уровней защищаемой системы в целом. Сформированные множества вариантов построения подсистемы, входящей в состав системы защиты от компьютерных атак, для каждого уровня защищаемой системы в отдельности будут иметь вид $V_{kl} = \{S_{k1}, S_{k2}, \dots, S_{kN}\}$, где k – подсистема, входящая в систему защиты от компьютерных атак; l – уровень построения защищаемой системы; N – количество всех возможных вариантов построения k -й подсистемы для l -го уровня защищаемой системы ($l = \overline{1, L}$, где L – количество уровней в защищаемой системе).

Формирование множества вариантов V_k построения k -й подсистемы, входящей в систему защиты от компьютерных атак, для всех уровней защищаемой системы в целом осуществляется путем прямого произведения множеств всех возможных ва-



риантов построения k -й подсистемы для каждого l -го уровня защищаемой системы [9]: $V_k = \prod_{l=1}^L V_{kl}$.

Для подсистемы обнаружения компьютерных атак данное выражение будет выглядеть следующим образом: $V_{обн} = \prod_{l=1}^L V_{обнl}$.

Для подсистемы противодействия компьютерным атакам оно приобретет вид $V_{пр} = \prod_{l=1}^L V_{прl}$.

Для подсистемы устранения последствий применения компьютерных атак выражение будет следующим: $V_{устр} = \prod_{l=1}^L V_{устрl}$.

Отбор вариантов построения подсистем для конкретной конфигурации защищаемой информационно-вычислительной (автоматизированной) системы

Данная процедура осуществляется путем проверки совместимости компонентов подсистем системы защиты с программно-аппаратной плат-

формой, на базе которой построена защищаемая информационно-вычислительная (автоматизированная) система, компонентов между собой в пределах отдельной подсистемы одного уровня, компонентов подсистем разных уровней между собой, а также компонентов, входящих в состав одной подсистемы всех уровней, с компонентами, входящими в состав других подсистем всех уровней.

Совместимость компонентов, на которых построены подсистемы системы защиты с программно-аппаратной платформой, определяется путем проверки отсутствия конфликтов при обращении к программным и аппаратным ресурсам информационно-вычислительной (автоматизированной) системы (функции ядра операционной системы, системные области памяти, порты ввода-вывода и т.п.).

Совместимость компонентов между собой в пределах отдельной подсистемы одного уровня определяется исходя из возможности выполнения всех функциональных требований для данной подсистемы при совместной работе компонентов, входящих в состав этой подсистемы. Если какой-либо компонент блокирует выполнение какой-либо функции другим компонентом и при этом блокирование данной функции влечет за собой невыполнение функциональных требований, определенных для данной подсистемы, эти два компонента считаются несовместимыми в пределах подсистемы и данный вариант построения подсистемы в формируемое множество альтернативных вариантов построения подсистем не включается. Если какой-либо компонент блокирует выполнение какой-либо функции другим компонентом, но при этом все функциональные требования, определенные для подсистемы, выполняются, данный вариант построения подсистемы включается во множество альтернативных вариантов.

Совместимость компонентов подсистем, входящих в состав системы защиты разных уровней, между собой и компонентов, входящих в состав одной подсистемы всех уровней, с компонентами других подсистем всех уровней также проверяется исходя из возможности выполнения всех функциональных требований, определенных для отдельных подсистем, входящих в систему защиты от компьютерных атак.

Предложенный в статье подход позволяет формировать множество возможных вариантов построения системы защиты от компьютерных атак с учетом выполнения всех функциональных требований, возложенных на подсистемы, входящие в состав системы защиты, с учетом особенностей функционирования и построения защищаемой системы, а также с учетом декомпозиции системы защиты от компьютерных атак на три подсистемы: обнаружения компьютерных атак, противодействия компьютерным атакам и устранения последствий применения компьютерных атак.

Литература

1. Дроботун Е.Б. Синтез систем защиты автоматизированных систем управления от разрушающих программных воздействий // Программные продукты и системы. 2016. Т. 29. № 3. С. 51–59.
2. Федоров Ю.Н. Справочник инженера по АСУ ТП: Проектирование и разработка. М.: Инфра-Инженерия, 2008. 928 с.
3. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах. М.: Горячая линия–Телеком, 2001. 148 с.
4. Основы проектирования и эксплуатации автоматизированных систем управления военного назначения; [под ред. В.Л. Ляковского]. М.: Изд-во МГТУ им. Н.Э. Баумана, 2016. 188 с.
5. Проскурин В.Г., Крутов С.В., Мацевич И.В. Программно-аппаратные средства обеспечения информационной безо-

пасности. Защита в операционных системах. М.: Радио и связь, 2000. 168 с.

6. Сердюк В.А. Организация и технология защиты информации: обнаружение и предотвращение информационных атак в автоматизированных системах предприятий. М.: Изд-во ВВШЭ, 2011. 572 с.
7. Шумский А.А., Шелупанов А.А. Системный анализ в защите информации. М.: Гелиос АРВ, 2005. 224 с.
8. Бородакий Ю.В., Добродеев А.Ю., Нашекин П.А., Бутусов И.В. Основной объект воздействия противника // Воздушно-космическая оборона. 2014. № 2. С. 30–37.
9. Осипова В.А. Основы дискретной математики. М.: ФОРУМ: ИНФРА-М, 2006. 160 с.
10. Поляничко М.А. Архитектура системы автоматизированного обнаружения и разрешения конфликтов программных средств защиты информации // Изв. ПГУПС. 2013. № 1. С. 39–45.

Software & Systems

DOI: 10.15827/0236-235X.118.314-319

Received 02.10.16

2017, vol. 30, no. 2, pp. 314–319

METHOD OF GENERATING SETS OF ALTERNATIVE VARIANTS OF BUILDING SUBSYSTEMS WHICH ARE A PART OF A COMPUTER ATTACK PROTECTION SYSTEM

E.B. Drobotun¹, Ph.D. (Engineering), Doctoral Student, drobotun@xakep.ru

E.P. Uglovsky¹, Head of Laboratory

I.Sh. Zamaltdinov¹, Ph.D. (Engineering), Senior Researcher

¹ Military Academy of the Aerospace Defense, Zhigareva St. 50, Tver, 170100, Russian Federation

Abstract. Constructing a rational computer attack protection system for an information or automated system assumes creating a set of protection system configurations consisting of a set of separate program and hardware-software components and a further choice of a rational option of creating a computer attack protection system from the created set according to certain criteria. When generating this set, in addition to protection system compliance with necessary functional requirements, it is necessary to consider the parameters of the protected system (its structure and a multiple-level creation system), as well as software and hardware compatibility of components, and also compatibility of components with a hardware-software platform, which is the basis for the constructed protected system.

The article presents one of the possible approaches to forming a set of possible options of creating a computer attack protection system, taking into account its decomposition on three subsystem types. They are: computer attack detection subsystems, computer attack counteraction subsystems and subsystems of elimination of consequences of computer attack application.

Keywords: automated control system, computer attack, information security, protection system design, computer attack protection.

References

1. Drobotun E.B. Synthesis of protection systems of automated control systems against destroying program influence. *Programmnye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 3, pp. 51–59 (in Russ.).
2. Fedorov Yu.N. *Spravochnik inzhenera po ASU TP: Proektirovaniye i razrabotka* [Automated Process Control System Engineer's Reference Book: Design and Development]. Moscow, Infra-Inzheneriya Publ., 2008, 928 p.
3. Maluk A.A., Pazizin S.V., Pogozhin N.S. *Vvedeniye v zashchitu informatsii v avtomatizirovannykh sistemakh* [Introduction to Information Protection in Automated Systems]. Moscow, Goryachaya liniya–Telekom Publ., 2001, 148 p. (in Russ.).
4. Lyaskovsky V.L. (Ed.) *Osnovy proektirovaniya i ekspluatatsii avtomatizirovannykh sistem voennogo naznacheniya* [Fundamentals of Design and Operation of Military Automated Control Systems]. Study guide. Moscow, N.E. Bauman MSTU Univ. Publ., 188 p.
5. Proskurin V.G., Krutov S.V., Matsevich I.V. *Programmno-apparatnye sredstva obecpecheniya bezopasnosti. Zashchita v operatsionnykh sistemakh* [Hardware-software Means of Ensuring Information Security. Protection in Operating Systems]. Study guide. Moscow, Radio i svyaz Publ., 2000, 168 p.
6. Serduk V.A. *Organizatsiya i tekhnologiya zashchity informatsii: obnaruzhenie i predotvrashchenie informatsionnykh atak v avtomatizirovannykh sistemakh predpriyatiy* [Information Protection Organization and Technology: Detection and Prevention of Information Attacks in Enterprise Automated Systems]. Moscow, HSE Publ., 2011, 572 p.
7. Shumsky A.A., Shelupanov A.A. *Sistemnyy analiz v zashchite informatsii* [System Analysis in Information Security]. Moscow, Gelios ARV Publ., 2005, 224 p.
8. Borodakiy Yu.V., Dobtodeev A.Yu., Nashchekin P.A., Butusov I.V. The Main Object of Enemy's Influence. *Vozdushno-kosmicheskaya oborona* [Aerospace Defense]. 2014, no. 2, pp. 30–37 (in Russ.).
9. Osipova V.A. *Osnovy diskretnoy matematiki* [Fundamentals of Discrete Mathematics]. Study guide. Moscow, FORUM: INFRA-M Publ., 2006, 160 p.
10. Polyanchko M.A. Architecture of the system of detection and resolution of the software security conflicts. *Izvestiya PGUPS* [Proc. of Petersburg Transport Univ.]. 2013, no. 1(34), pp. 39–45 (in Russ.).

УДК 378.046.4+349

DOI: 10.15827/0236-235X.118.320-323

Дата подачи статьи: 30.01.17

2017. Т. 30. № 2. С. 320–323

УРОВНИ И ПРАВОВАЯ МОДЕЛЬ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ (ЗАЩИТЫ ИНФОРМАЦИИ)

С.В. Голубчиков, к.т.н., начальник отдела, gsv_64@list.ru

(ПАО «НПО «Алмаз», Ленинградский просп., 80, стр. 16, г. Москва, 125190, Россия);

В.К. Новиков, к.в.н., доцент

(Военная академия РВСН им. Петра Великого,

ул. Карбышева, 8, г. Балашиха, Московская обл., 143900, Россия);

А.В. Баранова, аспирант, abv92@list.ru

(МГИМО (университет) МИД России, просп. Вернадского, 76, г. Москва, Россия)

В контексте определения информации как стратегического ресурса любого государства, производительной силы и дорогого товара рассматриваются проблемы информационной безопасности (защиты информации). Для их разрешения одним из направлений деятельности государства является правовое регулирование.

Так как информационная безопасность – это неотъемлемая часть общей и национальной безопасности, содержание которой базируется прежде всего на Конституции Российской Федерации, а также на основных базовых документах, в работе выделены уровни безопасности, дано понятие жизненно важных интересов, вытекающее из понятия безопасности.

Формулируются важнейшие задачи обеспечения информационной безопасности Российской Федерации. Задаются направления обеспечения информационной безопасности, а также организационно-технические мероприятия по защите информации в общегосударственных информационных и телекоммуникационных системах.

Предлагается к рассмотрению разработанная правовая модель обеспечения информационной безопасности, где выделены объекты защиты информации: персональные данные человека, различные технические средства, ПО, информационно-технические системы, документы и др.

В работе делается вывод о том, что информационная безопасность является составной частью общей и национальной безопасности и охватывает все сферы деятельности.

Ключевые слова: информация, информационная безопасность, защита информации, информационная сфера, правовая модель.

Начало XXI века ознаменовано бурным развитием информационных технологий во всех сферах государственной деятельности и общественной жизни. Информация все в большей мере становится стратегическим ресурсом любого государства, производительной силой и дорогим товаром. Это не может не вызывать стремления государств, организаций и отдельных граждан получить преимущества за счет завладения информацией, нанесения ущерба информационным ресурсам конкурента, а также защиты своих информационных ресурсов [1–3].

В настоящее время вопрос информационной безопасности остро стоит на уровне как государства, различных организаций, так и отдельных граждан. Важно обеспечить их конституционные права на получение достоверной информации, на ее использование в интересах осуществления законной деятельности учреждений, а также на защиту государственной, коммерческой, семейной, личной и других видов тайн.

Рассмотрим систему информационной безопасности, сложившуюся в России.

Проблема защиты информации от постороннего доступа и нежелательных воздействий возникла с развитием общественных отношений. Наиболее ценной становится информация, позволяющая ее владельцу получить какой-либо материальный, политический, военный и другой выигрыш [4–6].

Однако создание индустрии переработки информации порождает целый ряд сложных проблем, одной из которых является надежное обеспечение сохранности и установленного правового статуса информации в отдельных технических средствах, в информационно-вычислительных системах и информационно-телекоммуникационных сетях. Данная проблема вошла в обиход как проблема информационной безопасности [7–8].

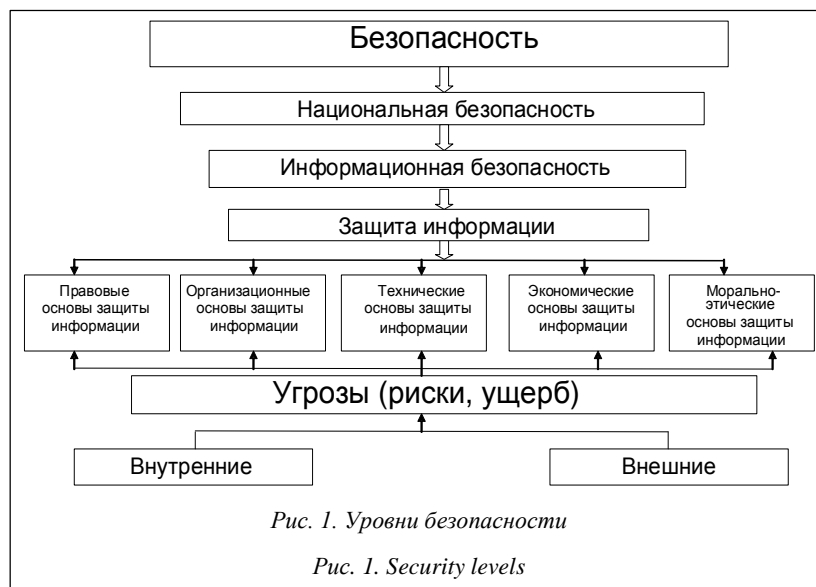
Для ее разрешения необходимо урегулировать на государственном уровне правовые аспекты данной области.

Информационная безопасность является неотъемлемой составной частью общей и национальной безопасности, содержание которой базируется в первую очередь на Конституции России, а также на основных базовых документах: Федеральном законе «О безопасности», «Стратегии национальной безопасности Российской Федерации до 2020 года» и Доктрине информационной безопасности Российской Федерации.

Анализ приведенных документов позволяет выделить уровни безопасности, представленные на рисунке 1.

Под безопасностью понимается состояние защищенности жизненно важных интересов личности, общества и государства от внутренних и внешних угроз.

Из понятия «безопасность» вытекает другое понятие – «жизненно важные интересы» как совокуп-



ность потребностей, удовлетворение которых надежно обеспечивает существование и возможности прогрессивного развития личности, общества и государства.

Рассмотрим суть информационной безопасности через понимание национальной безопасности.

Национальная безопасность – состояние защищенности личности, общества и государства от внутренних и внешних угроз, которое позволяет обеспечить конституционные права, свободы, достойные качество и уровень жизни граждан, суверенитет, территориальную целостность и устойчивое развитие Российской Федерации, оборону и безопасность государства.

При этом в рамках национальной безопасности национальные интересы России в информационной сфере заключаются в соблюдении конституционных прав и свобод в области получения информации и пользования ею, в развитии современных телекоммуникационных технологий, в защите государственных информационных ресурсов от несанкционированного доступа.

Угрозами национальной безопасности России в информационной сфере, представляющими серьезную опасность, являются:

- стремление ряда стран к доминированию в мировом информационном пространстве;
- вытеснение России с внешнего и внутреннего информационных рынков;
- разработка рядом государств концепций информационных войн;
- возможность нарушения нормального функционирования информационных и телекоммуникационных систем, получения несанкционированного доступа к ним.

Важнейшие задачи обеспечения информационной безопасности:

- реализация конституционных прав и свобод граждан в сфере информационной деятельности;

- совершенствование и защита отечественной информационной инфраструктуры, интеграция России в мировое информационное пространство;

- противодействие угрозе развязывания противоборства в информационной сфере.

Направлениями обеспечения информационной безопасности [9] Российской Федерации в общегосударственных информационных и телекоммуникационных системах являются:

- предотвращение перехвата информации из помещений и с объектов, а также информации, передаваемой по каналам связи с помощью технических средств;

- исключение несанкционированного доступа к обрабатываемой или хранящейся в технических средствах информации;

- предотвращение утечки информации по техническим каналам, возникающей при эксплуатации технических средств ее обработки, хранения и передачи;

- предотвращение специальных программно-технических воздействий, вызывающих разрушение, уничтожение, искажение информации или сбой в работе средств информатизации;

- обеспечение информационной безопасности при подключении общегосударственных информационных и телекоммуникационных систем к внешним информационным сетям, включая международные;

- обеспечение безопасности конфиденциальной информации при взаимодействии информационных и телекоммуникационных систем различных классов защищенности;

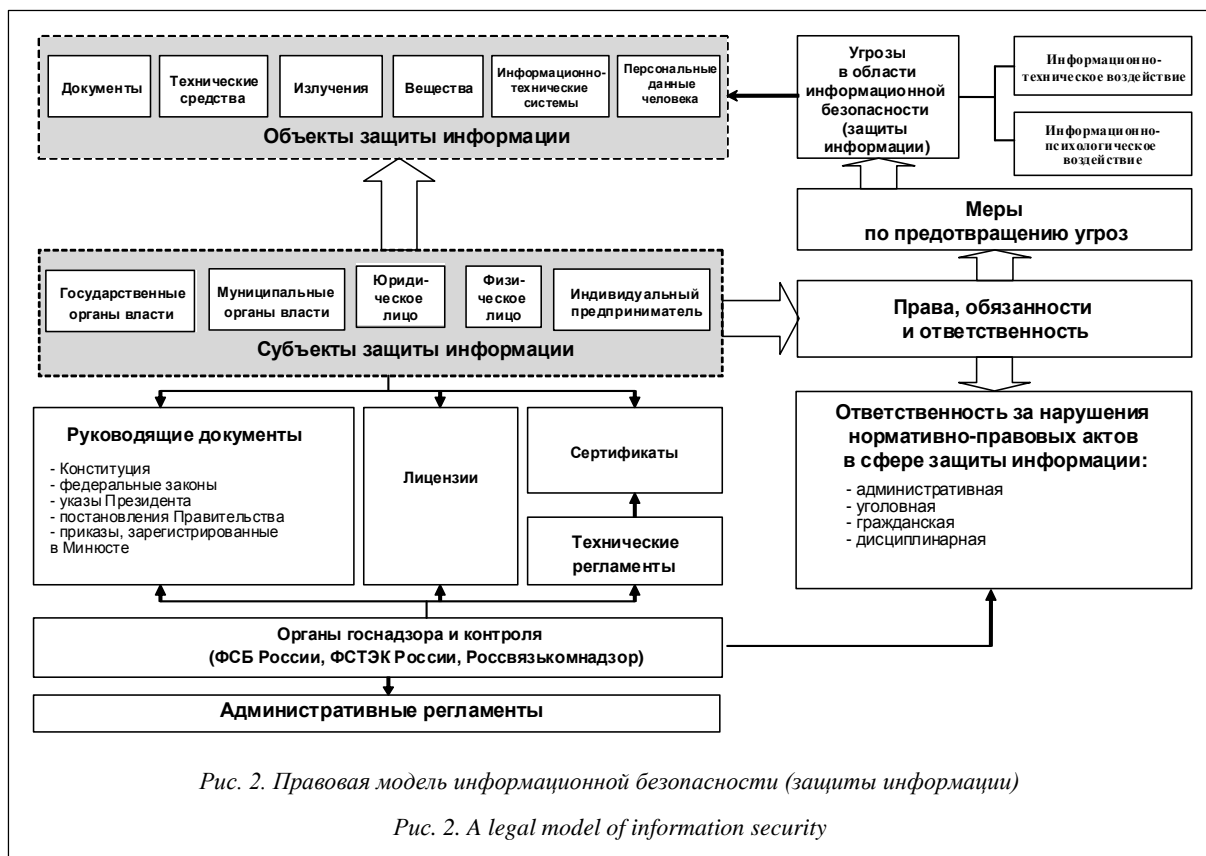
- выявление внедренных на объекты и в технические средства электронных устройств перехвата информации.

Защитить информацию в общегосударственных информационных и телекоммуникационных системах позволяют организационно-технические мероприятия:

- лицензирование деятельности организаций в области защиты информации;

- аттестация объектов информатизации по выполнению требований обеспечения защиты информации при проведении работ, связанных с использованием сведений, составляющих государственную тайну;

- сертификация средств защиты информации и контроля эффективности их использования, а также защищенности информации от утечки по техническим каналам систем и средств информатизации и связи;



– введение территориальных, частотных, энергетических, пространственных и временных ограничений в режимах использования технических средств, подлежащих защите;

– создание и применение информационных и автоматизированных систем управления в защищенном исполнении.

Анализ сущности и содержания нормативных правовых актов Российской Федерации в области информационной безопасности (защиты информации) позволил сформулировать правовую модель информационной безопасности (защиты информации) [10], которая дает более полное понятие правовой основы защиты информации (рис. 2).

Правовая модель информационной безопасности (защиты информации) позволяет представить область деятельности организаций и ее участников, правовые отношения, возникающие при восприятии (сборе), обработке, хранении, передаче и уничтожении и других действиях, производимых с информацией.

Таким образом, информационная безопасность является составной частью общей и национальной безопасности и охватывает все сферы деятельности государства, гражданина, а также различных организаций и бизнеса [11, 12].

Литература

1. Буданцев Ю.П. Информационное оружие и информационная война в современных условиях // Безопасность. 1999. № 3–4. С. 103–115.

2. Манилов В.Л. Национальная безопасность: ценности, интересы, цели // Военная мысль. 2005. № 9. С. 7–8.

3. Мухин В., Лось А., Новиков В. Информационная война в Персидском заливе // Безопасность. 1996. № 1–2. С. 60–63.

4. Новопашина А.П. Информационная политика России: состояние и пути совершенствования // Безопасность. 2010. № 11–12. С. 5–23.

5. Панарин И.Н., Панарина Л.Г. Информационная война и мир. М.: ОЛМА-ПРЕСС, 2003. 384 с.

6. Кириленко В.И., Лось В.П. Информационная борьба и проблемы обеспечения информационной безопасности Российской Федерации // Информационное право. Информационная культура и информационная безопасность: матер Всерос. науч.-практич. конф. СПб: Изд-во ГУП, 2002. С. 59–63.

7. Киреев А.Е. Современные проблемы в области информационной безопасности: классические угрозы, методы и средства их предотвращения // Молодой ученый. 2012. № 3. С. 40–45.

8. Петров В.П., Петров С.В. Информационная безопасность человека и общества. М.: ЭНАС, 2007. 336 с.

9. Новиков В.К., Галушкин И.Б., Аксенов С.В. Информационная безопасность и защита информации. Организационно-правовые основы. М.: Горячая линия–Телеком, 2016. 312 с.

10. Голубчиков С.В., Новиков В.К., Баранова А.В. Правовая модель информационной безопасности (защиты информации) в образовательном процессе // Муниципальное образование: инновации и эксперимент. 2017. № 1. С. 55–58.

11. Новиков В.К. Организационное и правовое обеспечение информационной безопасности. Ч. I. Правовое обеспечение информационной безопасности. М.: Изд-во МИЭТ, 2013. 184 с.

12. Новиков В.К., Галушкин И.Б., Аксенов С.В. Организационно-правовые основы информационной безопасности (защиты информации). Ч. II. Правовые и организационные основы информационной безопасности (защиты информации). М.: Изд-во ВА РВСН им. Петра Великого, 2015. 395 с.

A SYSTEM AND LEGAL MODEL OF INFORMATION SECURITY (DATA PROTECTION)

S.V. Golubchikov¹, Ph.D. (Engineering), Head of Department, gsv_64@list.ru

V.K. Novikov², Ph.D. (Military Sciences), Associate Professor

A.V. Baranova³, Postgraduate Student, abv92@list.ru

¹ PJSC NPO "Almaz", Leningradsky Ave. 80/16, Moscow, 125190, Russian Federation

² The Military Academy of Strategic Rocket Troops after Peter the Great,
Karbysheva St. 8, Balashikha, Moscow Reg., 143900, Russian Federation

³ MGIMO University, Vernadsky Ave. 76, Moscow, Russian Federation

Abstract. The paper considers the problems of information security (data protection) in the context of information as a strategic resource of any state, a productive force and an expensive commodity. One of the state activities to solve these problems is the legal regulation of this area.

Since information security is an integral part of the overall and national security with its content based on the Constitution of the Russian Federation, as well as on the core documents, the article identifies the levels of security of the Russian Federation.

Security is defined as a condition of protection of the vital interests of an individual, society and the state from internal and external threats. Therefore the paper gives the definition of "vital interests" arising from the definition of "security".

The authors formulate the most important purposes for ensuring information security of the Russian Federation.

The paper specifies the directions of ensuring the information security of the Russian Federation, as well as organizational and technical measures to protect information in the national information and telecommunication systems.

The article presents the developed legal model for ensuring information security. The model reveals the objects of information security that may be under different kinds of influences, such as personal data; various technical means; software; information and technical systems involved in sensing, processing, storage and transmission of information (data); documents (paper, electronic); radiation; substances.

There is a conclusion that information security is an integral part of the overall and national security and covers all spheres of activities of the state, citizens, as well as various organizations and businesses.

Keywords: information, information security, data protection, information sphere, legal model.

References

1. Budantsev Yu.P. Information weapon and information war under current conditions. *Bezopasnost* [Security]. 1999, no. 3–4, pp. 23–29 (in Russ.), pp. 103–115.
2. Manilov V.L. National security: values, interests, purposes. *Voennaya mysl* [Military Thought]. 2005, no. 9, pp. 7–8 (in Russ.).
3. Mukhin V., Los A., Novikov V. Information war in the Persian Gulf. *Bezopasnost* [Security]. 1996, no. 1–2, pp. 60–63 (in Russ.).
4. Novopashin A.P. Information policy of Russia: status and ways to improve. *Bezopasnost* [Security]. 2010, no. 11–12, pp. 5–23 (in Russ.).
5. Panarin I.N., Panarina L.G. *Informatsionnaya voyna i mir* [Information War and Peace]. Moscow, OLMA-PRESS, 2003, 135 p.
6. Kirilenko V.I., Los V.P. Information war and information security problems in the Russian Federation. *Informatsionnoe pravo. Informatsionnaya kultura i informatsionnaya bezopasnost: mater. Vseros. nauch.-praktich. konf.* [Information Right. Information Culture and Information Security: Proc. All-Russian Science and Practice Conf.]. St. Petersburg, GUP Publ., 2002, pp. 59–63 (in Russ.).
7. Kireenko A.E. Modern problems in information security: classical threats, preventing methods and instruments. *Molodoy ucheny* [Young Scientist]. 2012, no. 3 (38), pp. 40–45 (in Russ.).
8. Petrov V.P., Petrov S.V. *Informatsionnaya bezopasnost cheloveka i obshchestva* [Information Security for a Human and Society]. ENAS Publ., 2007.
9. Novikov V.K., Galushkin I.B., Aksenov S.V. *Informatsionnaya bezopasnost i zashchita informatsii. Organizatsionno-pravovye osnovy* [Information Security and Protection. Procedural and Institutional Basics]. V.K. Novikov (Ed.). Moscow, Goryachaya liniya–Telekom Publ., 2016, 312 p.
10. Golubchikov S.V., Novikov V.K., Baranova A.V. Information Security Legal Model in an Educational Process. *Munitsipalnoe obrazovanie: innovatsii i eksperiment* [Municipal Entity: Innovations and Experiments]. 2017, no. 1, pp. 55–58 (in Russ.).
11. Novikov V.K. *Organizatsionnoe i pravovoe obespechenie informatsionnoy bezopasnosti. Ch. I. Pravovoe obespechenie informatsionnoy bezopasnosti* [Procedural and Institutional Support for Information Security. Part 1. Legal Support for Information Security]. Moscow, MIET Publ., 2013, 184 p.
12. Novikov V.K., Galushkin I.B., Aksenov S.V. *Organizatsionno-pravovye osnovy informatsionnoy bezopasnosti (zashchity informatsii). Ch. II. Pravovye i organizatsionnye osnovy informatsionnoy bezopasnosti (zashchity informatsii)* [Procedural and Institutional Basics of Information Security. Part 2. Legal and Institutional Basis of Information Security]. Moscow, VA RVSN im. Petra Velikogo Publ., 2015, 395 p.

УДК 004.056.53
DOI: 10.15827/0236-235X.33: .324-327

Дата подачи статьи: 25.11.16
2017. Т. 30. № 2. С. 324–327

ИССЛЕДОВАНИЕ ЗАЩИЩЕННОСТИ СИСТЕМ ЭЛЕКТРОННЫХ ПЛАТЕЖЕЙ

Т.А. Маркина, к.т.н., доцент, tmark812@mail.ru;

В.А. Хрупов, студент, od@mail.ifmo.ru

(Университет ИТМО, Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)

В данной работе исследуется защищенность систем электронных платежей. Актуальность исследования обусловлена постоянным ростом количества преступлений в сфере электронных платежей, что подтверждается официальной статистикой. Пользователям систем электронных платежей необходимо иметь представление о защищенности этих систем, следовательно, и о защищенности своих денежных средств, хранящихся в них.

В процессе работы рассматривались разнообразные системы электронных платежей, проанализированы способы и методы, обеспечивающие безопасность систем, выявлены недостатки и достоинства способов и методов их защиты, определены критерии и подкритерии оценки защищенности, а для каждого из подкритериев определено значение показателя безопасности в процентах.

На основании значений показателей была произведена оценка защищенности систем электронных платежей, что позволяет сделать выводы о степени защищенности каждой из них.

Исследование позволяет дать оценку защищенности наиболее популярных систем электронных платежей на территории Российской Федерации и стран СНГ.

Ключевые слова: система электронных платежей, электронный платеж, электронные деньги, защищенность системы электронных платежей, мобильный сервис системы электронных платежей, немобильный сервис системы электронных платежей, карточная платежная система.

Информатизация современного общества несет в себе как позитивное, так и негативное начало. В связи с массовой компьютеризацией информационных процессов, увеличением ценности и значимости информационных ресурсов перед обществом встает острая проблема защиты информации и персонализированных данных, которые циркулируют в важных информационных системах, то есть предотвращение модифицирования, удаления и похищения для использования в криминальных целях.

Система электронных платежей (СЭП) – это технология (если говорить о реализации, то сервис), представляющая собой совокупность методов, договоренностей и подтехнологий и позволяющая производить расчеты между контрагентами по сетям передачи данных [1]. СЭП входят в сферу электронной коммерции. В настоящее время главной угрозой для электронной коммерции являются преступления в сфере информационных технологий, или киберпреступность [2]. Противоправные действия в области систем электронных платежей могут повлечь за собой нарушения в работе ЭВМ и привести СЭП, банк, а впоследствии и рыночную экономику к краху. Проблемами защиты данных в СЭП, в том числе электронных платежей, разработкой современных технических и клиентских средств защиты, а также усовершенствованием антивирусных систем сегодня занимаются многие отечественные и иностранные специалисты (см. [3, 4] и др.), а также крупные компании: Eset Software Ltd, Kaspersky AntiVirus [5] и многие другие. Существует значительное количество научных исследований и практических разработок, связанных с защитой СЭП, однако в силу постоянного разви-

тия технологий возникают новые проблемы, которые необходимо решать новыми способами [6].

Под защищенностью СЭП понимается совокупность технических и клиентских средств защиты, обеспечивающих требуемый уровень безопасности. В настоящий момент средства криптографической и технической защиты еще не имеют длительной истории успешной эксплуатации и безопасность (подделка, хищение, изменение номинала) не подтверждена широким обращением [2]. Теоретически возможны хищения электронных денег посредством инновационных методов в силу недостаточной зрелости технологий защиты [4]. Актуальность данного исследования подтверждена официальной статистикой. По данным МВД, в 2013 году зарегистрировано более 11 тысяч преступлений в сфере электронных платежей, а к середине 2014 года – более 7 тысяч [7]. По приведенной статистике можно судить о том, что количество преступлений в сфере электронных платежей растет с каждым годом. Это означает, что операторы СЭП не торопятся создавать и вводить новые методы и способы защиты платежей, а пользователям необходимо самим выбирать наиболее безопасные СЭП для оплаты товаров и услуг.

СЭП подразделяются на мобильные сервисы (приложения СЭП для мобильных устройств) и немобильные (браузерные СЭП, приложения СЭП для ПК). По видам оплаты СЭП подразделяют на карточные платежные системы и операторы электронных денег.

Платежные шлюзы – симбиоз карточных платежных систем и операторов цифровой наличности. Такие системы позволяют совершать практически все возможные транзакции в рамках одной

Таблица 1

Критерии и подкритерии безопасности (их особенности)

Table 1

Safety criteria and sub-criteria (their features)

Критерии и подкритерии безопасности (их особенности)		Значение показателя безопасности, %
1) Первичная защита аккаунта СЭП		
1. Защита аккаунта паролем (критерий 1)		
Наличие пароля аккаунта		10
Надежность пароля	Минимальный пароль 1 символ	0
	Минимальный пароль 5-6 символов	5
	Минимальный пароль 8 символов	10
	Наличие доп. условий (спец. символы, буквы верхнего регистра, цифры)	5
Наличие строки надежности пароля		2
Ограничение срока действия пароля СЭП		3
2. Использование безопасного соединения с веб-сайтом (критерий 2)		
Безопасность SSL-соединения	Не используется SSL-шифрование	0
	Применяется SSL-шифрование, но есть незащищенный контент, представляющий серьезную опасность	3
	Применяется SSL-шифрование, но обнаружен незащищенный контент	5
	Используется SSL-шифрование (безопасное соединение)	10
Используемый протокол	С протоколом TLS 1.1	5
	С протоколом TLS 1.2	10
2) Безопасность при авторизации в СЭП		
Подтверждение входа с помощью мобильного телефона, сервиса E-num или e-mail (критерий 3)	Мобильный телефон	5
	Сервис E-num	5
	E-mail	5
3) Авторизация с помощью технических настроек		
1. Возможность ограничения доступа по IP-адресу (критерий 4)		5
2. Выдача персонального цифрового сертификата для доступа в СЭП (критерий 5)		5
4) Подтверждение операций с кошельком дополнительным паролем		
Подтверждение операций (критерий 6)	С помощью SMS присылаемое на моб. телефон или e-mail	5
	С помощью сервисов E-num, Google Authenticator или другими подобными системами	5
	С помощью дополнительного платежного пароля	5
5) Дополнительные способы и методы, обеспечивающие безопасность денежных средств		
Возможность привязки почты/телефона к СЭП дополнительно (критерий 7)		3
1. Возможность выпуска или приобретения виртуальной карты с коротким сроком действия и лимитом средств (критерий 8)		3
2. Наличие идентификации с подтверждением документами пользователя (критерий 9)		3
3. Использование секретных вопросов или секретного слова (критерий 10)		3
4. Ограничение сессии – автоматический выход из системы по истечении времени неактивности пользователя (критерий 11)		3
6) Информационные способы обеспечения безопасности		
1. Информирование пользователя по SMS или e-mail об операциях, проводимых с электронным кошельком (критерий 12)		3
2. Наличие журнала посещений пользователем СЭП (критерий 13)		3
3. Наличие инструкций и рекомендаций по безопасности для пользователей СЭП (критерий 14)		3
4. Наличие службы поддержки (критерий 15)	По телефону	3
	Через форму обратной связи или e-mail	2

системы. Большинство СЭП можно отнести к шлюзам, но обычно каждая СЭП выделяет основной тип электронных платежей [3].

В ходе работы был проведен анализ способов и методов защиты мобильных и немобильных сервисов СЭП, таких как протоколы SSL и SET [1], объединенный протокол SSL и SET, CVV2- и CVK2-код, проверка адреса Address Verification Service, однофакторная и двухфакторная авторизация, электронные сертификаты пользователя,

E-num, электронный токен, привязка IP, биометрические данные и др. [8].

Анализ защиты мобильных сервисов СЭП показал, что у большинства методов и способов защиты выявлены недостатки и уязвимости, которые дают возможность хищения денежных средств пользователя. Однако законодательные и технические меры по ограничению платежей не позволяют похитить с помощью мобильных сервисов СЭП крупные денежные средства.

Также был проведен анализ защиты немобильных сервисов СЭП. Он показал, что у немобильных сервисов существует большое количество разных методов защиты и только половина из них имеют недостатки и уязвимости. Это позволяет использовать немобильные сервисы как безопасное средство платежей, однако именно через них злоумышленники крадут крупные денежные средства [9]. Исследование проводилось на сервисах СЭП, в частности, тех, которые используют в обороте эмитированные электронные деньги и не относятся к интернет-банкингам.

В процессе работы были определены критерии и подкритерии оценки защищенности СЭП (табл. 1). Если какой-то метод или способ безопасности отсутствует, значение безопасности данного критерия будет равно 0 %. В сумме все критерии дают оценку безопасности 100 %. Оценка защищенности системы зависит от количества набранных процентов из 100. Оценка А (отлично) – от 90 % (включительно) и выше, оценка В (хорошо) – от 80 % (включительно) до 90 %, оценка С (удовлетворительно) – от 70 % (включительно) до 80 %, оценка F (неудовлетворительно) – результаты менее 70 %.

Всего выделено 15 критериев безопасности, они разбиты на 6 групп по степени безопасности. Например, в первой подгруппе первый критерий –

защита аккаунта паролем, где подкритериями выступают наличие пароля, надежность пароля (то есть минимальный пароль начинается от 1, 5, 8 символов, необходимо ли вводить дополнительные символы), наличие строки надежности пароля, ограничение срока действия пароля системой (например несколько месяцев). Вторым критерием в группе является использование безопасного соединения веб-сайтом, где подкритериями являются безопасность SSL-соединения (то есть используется ли SSL-шифрование и есть ли незащищенный контент на веб-странице), вторым подкритерием является используемый протокол (TLS 1.1 или 1.2, в котором не используются небезопасные алгоритмы шифрования).

Для исследования были взяты 12 систем электронных платежей, имеющих интерфейс на русском языке и находящихся в доменной зоне RU [10]. Это обусловлено тем, что именно эти системы под прицелом у злоумышленников из-за большого количества пользователей. Выбор этих систем немаловажен для рядовых пользователей, так как каждый человек хочет иметь наиболее защищенный инструмент электронных платежей. Результаты исследования представлены в таблице 2.

Исследование показало, что лишь две СЭП – WebMoney и OKPAY – имеют оценку «хорошо» (В). Всего три СЭП (VISA QIWI-кошелек, PayPal,

Таблица 2

Защищенность систем электронных платежей

Table 2

Security of electronic payment systems

№	СЭП	Критерий															
		Защита аккаунта СЭП паролем		Безопасность при авторизации в СЭП	Авторизация с помощью технических настроек		Подтверждение операций с кошельком доп. паролем	Дополнительные способы и методы						Информационные способы обеспечения безопасности			Оценка, сумма показателей
		1	2		3	4		5	6	7	8	9	10	11	12	13	
1	VISA QIWI-кошелек	27	20	0	0	0	5	3	3	3	0	3	3	0	3	6	C – 76 %
2	Яндекс. Деньги	18	15	0	0	0	5	3	0	3	3	0	3	3	3	6	F – 62 %
3	WebMoney	15	20	0	5	5	5	3	3	3	3	3	3	3	3	6	B – 80 %
4	PayPal	25	20	0	0	0	5	3	0	3	3	3	3	0	3	6	C – 74 %
5	RBK Money	18	20	0	5	0	5	3	0	3	3	3	3	0	3	6	C – 72 %
6	Rapida online	10	20	0	0	0	5	3	3	3	0	3	3	0	0	6	F – 56 %
7	Элекснет	10	8	0	0	0	5	3	0	3	3	3	3	0	0	6	F – 44 %
8	Wallet One	15	20	0	0	0	5	3	0	3	0	3	3	3	0	6	F – 61 %
9	Сомерау-кошелек	20	15	0	0	0	5	0	0	3	0	3	3	0	0	6	F – 55 %
10	Payeer-кошелек	20	15	5	0	0	5	3	0	3	3	3	3	3	0	6	F – 69 %
11	OKPAY	25	20	5	5	0	0	3	0	3	3	3	3	3	3	6	B – 82 %
12	MoneyMail	10	15	0	0	0	5	3	0	3	0	0	3	3	0	6	F – 48 %

РВК Money) получили оценку «удовлетворительно» (С). Все остальные СЭП получили оценку неудовлетворительно, среди них и Яндекс.Деньги. Для проведения электронных платежей можно рекомендовать использовать СЭП, получившие оценку «удовлетворительно» и выше, но при условии, что пользователь будет следовать инструкциям и рекомендациям СЭП, незамедлительно реагировать на подозрительные действия со своим электронным кошельком и сообщать об этом в службу поддержки СЭП, которую он использует.

Литература

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб: Питер, 2012. 960 с.
2. Иванов М.А., Михайлов Д.М., Чугунков И.В. Защита

информации в электронных платежных системах. М.: КНОРУС, 2011.

3. Голдовский И. Безопасность платежей в Интернете. СПб: Питер, 2005. 240 с.
4. Уэллс Т. Компьютерное мошенничество. Битва байтов. М.: Маросейка, 2010. 354 с.
5. Официальный веб-сайт лаборатории Касперского. URL: <http://www.kaspersky.com> (дата обращения: 15.07.16).
6. Ануреев С.В. Платежные системы и их развитие в России. М.: Финансы и статистика, 2005. 288 с.
7. Официальный веб-сайт делового портала TADVISER. URL: <http://www.tadviser.ru> (дата обращения: 15.07.16).
8. IT-сообщество Хабрахабр. URL: <http://habrahabr.ru> (дата обращения: 15.07.16).
9. Гризов А. Новые платежные технологии. Информационно-справочное издание. М.: Рекон Интернешнл, 2009. 468 с.
10. Официальный веб-сайт информационного агентства «Банки.ру». URL: <http://www.banki.ru> (дата обращения: 15.07.16).

Software & Systems

DOI: 10.15827/0236-235X.118.324-327

Received 25.11.16

2017, vol. 30, no. 2, pp. 324–327

INVESTIGATION OF SECURITY OF ELECTRONIC PAYMENT SYSTEMS

T.A. Markina¹, Ph.D (Engineering Sciences), Associate Professor, tmark812@mail.ru

V.A. Khrupov¹, Student, od@mail.ifmo.ru

¹ ITMO University, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. The article investigates security of electronic payment systems. The relevance of this study is confirmed by the official statistics, which shows the increase of the number of crimes in the field of electronic payments year by year. Therefore, users of these systems need to be aware of the security of electronic payment systems, consequently, about the security of their money stored in these systems.

The work considers a variety of electronic payment systems, analyzes the ways and methods of ensuring the security of electronic payment systems, identifies advantages and disadvantages of techniques and methods of electronic payment system protection, the criteria and sub-criteria for evaluation of electronic payment system security, and determines value security index in percentage terms for each sub-criteria.

Based on the values of the indicators the authors assessed security of electronic payment systems and received an overall assessment for each electronic payment system. Based on the assessment, the paper shows some conclusions about the degree of security of an electronic payment system.

This study demonstrates the assessment of security of the most popular electronic payment systems in the Russian Federation and CIS countries.

Keywords: electronic payment systems, electronic payment, e-money, security, electronic payment systems, mobile service electronic payment system, no mobile service electronic payment systems, card payment system.

References

1. Tanenbaum A., Wetherall D. *Computer Networks*. 5th ed., Prentice Hall Publ., 2010, 962 p.
2. Ivanov M.A., Mikhaylov D.M., Chugunkov I.V. *Zashchita informatsii v elektronnykh platezhnykh sistemakh* [Data Protection in Electronic Payment Systems]. Moscow, KNORUS Publ., 2011.
3. Goldovsky I. *Bezopasnost platezhey v Internetе* [Secure Payments on the Internet]. St. Petersburg, Piter Publ., 2005, 240 p.
4. Wells J.T. *Computer Fraud Casebook: The Bytes that Bite*. Wiley Publ., 2009, 440 p. (Russ. ed.: Moscow, Maroseyka Publ., 2010, 354 p.).
5. *Kaspersky Lab*. Official website. Available at: <http://www.kaspersky.com> (accessed July 15, 2016).
6. Anureev S.V. *Platzhnye sistemy i ikh razvitie v Rossii* [Payment Systems and Their Development in Russia]. Moscow, Finansy i statistika Publ., 2005, 288 p.
7. *TADVISER*. Official website. Available at: <http://www.tadviser.ru> (accessed July 15, 2016).
8. *IT-community Habrahabr*. Available at: <http://habrahabr.ru> (accessed July 15, 2016).
9. Grizov A. *Novye platezhnye tekhnologii. Informatsionno-spravочное izdanie* [New Payment Technologies. Information and Reference Edition]. Moscow, Rekon Interneshnl Publ., 2009, 468 p.
10. *Banki.ru*. Official website. Available at: <http://www.banki.ru> (accessed July 15, 2016).

УДК 621.039

DOI: 10.15827/0236-235X.118.328-332

Дата подачи статьи: 07.11.16

2017. Т. 30. № 2. С. 328–332

**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИЗМЕРЕНИЯ ВРЕМЕНИ
АУДИО-МОТОРНЫХ РЕАКЦИЙ ОПЕРАТОРОВ СИСТЕМ УПРАВЛЕНИЯ
ХИМИКО-ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ**

О.А. Ахремчик, д.т.н., доцент; И.И. Базулев, аспирант, bazulevivan@gmail.com
(Тверской государственный технический университет,
наб. Аф. Никитина, 22, г. Тверь, 170026, Россия)

Время сенсорно-моторной реакции оператора является основой для выбора моделей представления информации и стратегии управления химико-технологическим процессом на основе минимизации потерь в ходе человеко-машинного взаимодействия. Широкое применение звуковых сигналов для формирования сообщений аварийной и предупредительной сигнализаций диктует необходимость разработки методов и средств контроля времени аудио-моторных реакций.

В статье рассматриваются структура, функциональные характеристики и основные составляющие программного комплекса для измерения времени простых аудио-моторных реакций оператора системы управления. Исходя из концепции структурного программирования для реализации комплекса выбираются средства среды Delphi. Описываются типы экранных форм программного комплекса и возможные действия оператора с их использованием.

В качестве настроечных параметров комплекса предлагается использовать тип звукового сигнала, интервал времени, определяющий момент предъявления сигнала, число предъявлений сигнала, а в качестве стимулов при измерении – тональные сигналы разной частоты. В ходе измерения заполняются и используются БД звуковых сигналов, операторов и результатов измерений.

Предполагается применение программного комплекса для работы в составе системы управления химико-технологическим процессом. Комплекс может рассматриваться как инструмент для косвенной оценки состояния центральной нервной системы оператора. Основой передачи результатов измерения другим программным системам является сохранение результатов в форматах *.cds и *.txt.

Результаты апробации комплекса показывают, что характер влияния совокупности внешних факторов на время аудио-моторной реакции оператора носит мультипликативный характер.

Ключевые слова: звуковой сигнал, время, оператор, программный комплекс, реакция.

Функционирование оператора в составе систем управления и контроля становится все более характерным для современного производства [1]. В подсистемах оповещения и сигнализации в АСУ химико-технологическими и биотехнологическими процессами широко используются звуковые сигналы. Их применение для формирования сообщений аварийной и предупредительной сигнализаций обусловлено тем, что время реакции человека на звуковой сигнал меньше времени реакции на тактильное раздражение и восприятие звука в меньшей степени зависит от положения головы оператора, чем при использовании зрительного анализатора. Кроме того, использование звуковых сигналов позволяет снизить информационную сложность панелей оператора.

Данные соображения являются предпосылкой для использования дополнительных обратных связей в АСУ, обуславливающих передачу в систему управления информации о характеристиках оператора как составляющего компонента системы. Время отклика оператора на изменения в системе управления и управляемом процессе может косвенно оцениваться по времени простых сенсорно-моторных реакций оператора на программно формируемые раздражители (в нашем случае – время простой аудио-моторной реакции при срабатывании звуковой сигнализации).

Метод регистрации сенсорно-моторных реакций благодаря простоте и информативности ис-

пользуется для изучения когнитивных процессов, оценки функционального состояния центральной нервной системы, сенсорной чувствительности [2]. Под временем сенсорно-моторной реакции понимают интервал времени между появлением сигнала и ответной реакцией, который определяется скоростями возбуждения рецептора и послышки возникшего импульса в чувствительный центр, переработки сигнала в центральной нервной системе, принятия решения о реагировании на сигнал, передачи сигнала к началу действия по эфферентным волокнам, развития возбуждения в мышце и преодоления инерции части тела [2].

Время реакции отражает уровень неспецифической и специфической активации центральной нервной системы оператора. Временные и точностные показатели сенсорно-моторной реакции являются относительно постоянным свойством человека на протяжении определенных отрезков онтогенеза и могут рассматриваться как информативные показатели функционального состояния центральной нервной системы [3]. Малое время реакции отражает высокое качество нейрональной активности в ассоциативных зонах префронтальной коры головного мозга оператора, ответственных не только за анализ сенсорных сигналов, но и за организацию моторных реакций в ответ на них. Желательным является контроль сохранения значения времени аудио-моторной реакции оператора АСУ в определенном онтогенезом и опытом опера-

тора диапазоне при заданном уровне акустического и информационного шумов. Этим обусловлена актуальность разработки программного комплекса для измерения времени аудио-моторных реакций операторов АСУ. Предполагается, что комплекс будет использоваться с ПО конкретной АСУ химико-технологическим процессом параллельно с выполнением оператором своих рабочих функций в штатном режиме.

Множество функциональных характеристик комплекса определяется выражением по методике, предложенной при исследовании психофизиологического состояния человека в процессе обучения: $F = \{X1, X2, X3, X4, X5, X6, X7\}$, где $X1$ – класс объекта и системы управления; $X2$ – возможность дистанционного контроля состояния оператора; $X3$ – возможность проведения тестирования индивидуально и в группе; $X4$ – наличие документации для установки, настройки комплекса и проведения измерений; $X5$ – характеристики тестов, по которым проводится тестирование, и рекомендации по их выбору; $X6$ – свойства аппаратных средств; $X7$ – характеристики и особенности программных приложений [4].

На основе анализа множества функциональных характеристик программный комплекс для контроля времени аудио-моторных реакций оператора можно определить как стационарный, диагностический, функционирующий в составе АСУ, используемый для непрерывного мониторинга состояния нервной системы оператора. Планируется, что косвенным положительным эффектом от внедрения рассматриваемого комплекса будет сокращение времени проектирования интерфейсов НМИ-систем. Постановка задачи контроля состояния центральной нервной системы разработчика программных приложений при использовании систем автоматизированного проектирования связана с необходимостью применения технологии использования системных библиотек при сокращении времени проектирования [5].

Комплекс создавался исходя из опыта проектирования и эксплуатации программных систем для медико-биологических исследований [6, 7]. Выполненные работы включали разработку методики тестирования для определения времени простой аудио-моторной реакции, выбор модели и построение БД звуковых сигналов (стимулов), проектирование программных модулей для измерения времени реакции, хранения результатов тестирования, вывода полученной информации на монитор и сохранения данных измерения в архивных файлах.

При раскрытии перечня координат, определяющих характеристику тестов ($X5$), полагаем, что измерение проводится на рабочем месте и требует не более 0,2 % рабочего времени, число фаз тестирования – две, количество тестов – один, отсутствует параллельное измерение электрофизиологических показателей, при возникновении на объекте управ-

ления нештатной ситуации процесс измерения блокируется.

Обобщенная структура программного комплекса (рис. 1) предусматривает наличие, помимо аппаратного и программного, организационного обеспечения в виде документации, описывающей методику измерения времени реакции (координаты $X4$ и $X5$ множества функциональных характеристик комплекса). В первой фазе тестирования выбирается стимул из БД звуковых сигналов, для которого по результатам измерения времени простой аудио-моторной реакции в серии последовательных измерений среднее время реакции минимально. Первая фаза выполняется однократно перед началом производственного цикла или после перехода объекта и системы управления из нештатного в обычный режим функционирования. Вторая фаза тестирования связана с периодическим измерением времени реакции в ходе работы оператора при использовании определенного в ходе первой фазы типа стимула. Выбор периода измерения определяется свойствами последовательности средних значений времени реакции, получаемых в каждой серии измерений.

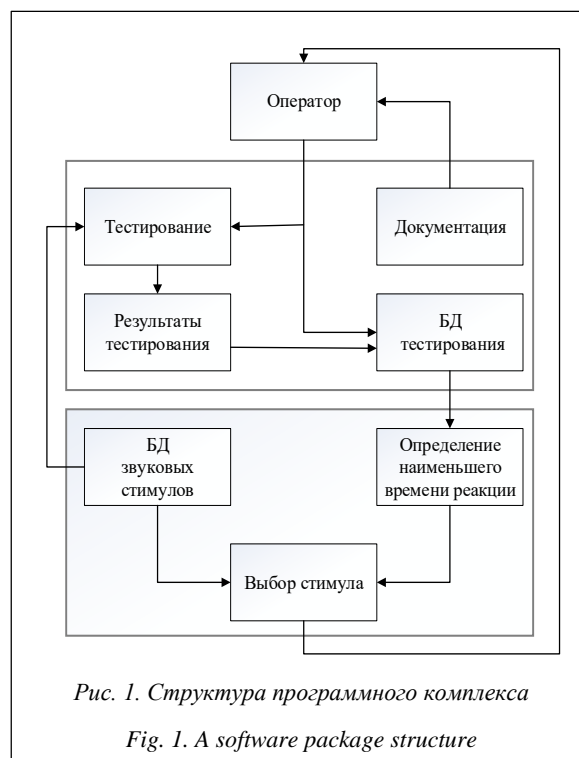


Рис. 1. Структура программного комплекса

Fig. 1. A software package structure

Частота и амплитуда звукового давления стимула настраиваются оператором при выборе типа стимула. В качестве стимула используется тональный сигнал из множества сигналов БД звуковых стимулов с частотой 300, 500, 1 000 Гц. При выборе частоты учитываются требования ГОСТ 21786-76, согласно которым в качестве сигналов аварийной сигнализации используются сигналы частотой от 800 до 5 000 Гц с звуковым давлением 90–100 дБ,

а для сигналов предупредительной сигнализации применяются сигналы с частотой 400–800 Гц и звуковым давлением 80–90 дБ.

Звуковой стимул может возникать либо в устройстве акустической сигнализации АСУ, либо в головных телефонах, которыми комплектуется компьютер оператора. Момент предъявления стимула задается с использованием моделирования псевдослучайной последовательности с равномерным законом распределения [8].

При сравнительном анализе сред Delphi фирмы Borland и Visual Studio фирмы Microsoft для разработки комплекса выбор сделан в пользу Delphi, которая поддерживает концепцию структурного программирования на встроенном в нее языке Object Pascal. Выбор был обусловлен наличием полного набора типов данных и возможностью автоматического создания проекта, включающего в себя файлы разных типов.

Требования к аппаратному обеспечению программного комплекса предусматривают наличие компьютера на базе Pentium I 100 МГц с RAM 16 Мб, с разрешением экрана не менее 800 на 600 точек, с установленной звуковой картой, способной обеспечить воспроизведение файлов типа *.wav. Желательно наличие персонального компьютера на базе Pentium IV (Celeron) 1 700 МГц с RAM 256 Мб, с разрешением экрана 1 024 на 768 точек.

Для функционирования программного комплекса необходима установка одной из операционных систем Windows NT 4.0 SP5 и выше.

Для чтения и генерации файлов результатов измерения в формате *.xls необходимо ПО MS Office (Excel) 97 и выше.

На первые версии программного комплекса получены свидетельства на программы для ЭВМ.

В данных версиях используются 12 внутренних программных модулей и 1 встроенный в установщик внешний модуль. Все полученные данные результатов измерения архивируются в БД в текстовом файле с расширением *.cds с возможностью сохранения в виде *.txt и переноса данных в таблицы Excel.

Результат измерения кодируется в виде $T = \langle N, BP, BT \rangle$, где N – номер проводимого измерения; BP – время реакции тестируемого, мс; BT – время и дата проведения тестирования (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/2.jpg).

Настроечными параметрами комплекса при проведении измерения являются число предъявления стимула (число повторений) и границы интервала времени между последовательным появлением звуковых сигналов, в пределах которого случайным образом формируется время подачи акустического тонального сигнала оператору (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/3.jpg). Время подачи определяется на основе генератора псевдослучайных чисел с использованием равномерного закона распределения.

Результаты измерения отражаются в виде двух связанных таблиц, являющихся частью БД тестирования. Первая таблица отображает табельный номер оператора, Ф.И.О., дату рождения, пол, а вторая – время реакции и время измерения (см. http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/4.jpg).

Апробация и верификация программного комплекса показали, что средние значения времени простой аудио-моторной реакции оператора на описанные выше стимулы в одной серии измерения при повторении измерений ежедневно в течение месяца можно рассматривать как выборочную реализацию значений квазистационарного случай-

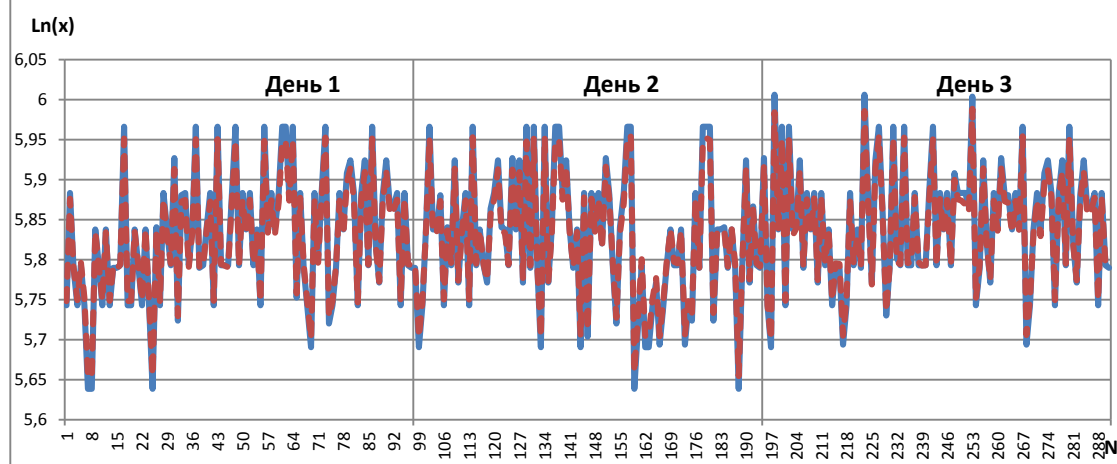


Рис. 2. Фрагмент выборочной реализации логарифма времени ($\ln(x)$) простой аудио-моторной реакции оператора при апробации программного комплекса

Fig. 2. A fragment of selected time log implementation ($\ln(x)$) of operator's simple audio motor reaction when testing a software package

ного процесса с логнормальным законом распределения (рис. 2). Это говорит о мультипликативном характере воздействия внешних факторов на время простой аудио-моторной реакции оператора АСУ химико-технологическим процессом. Причины и значимость учета вариабельности показателей времени реакции при исследованиях не рассматривались.

Для определения удобства работы оператора с комплексом при применении клавиатуры Microsoft Natural Ergonomic Keyboard 4000v1.0, встроенной аудиокарты GIGABYTE AMP-UP, головных телефонов Nama Insomnia H-51828 были проведены работы по оценке удобства пользования в соответствии с требованиями ГОСТ Р ИСО/МЭК 9126-93. В результате установлено, что интерфейс имеет положительную оценку удобства пользования.

В ходе работы с комплексом оператор открывает пять экранных меню (форм). В главном меню осуществляются переходы в меню База данных, Тестирование, Результаты тестирования, Информация о программе. Форма База данных предоставляет доступ к БД операторов и результатов измерения времени реакции оператора. Форма Тестирование предназначена для настройки параметров проведения теста. В ней выбирается тип звукового сигнала, устанавливаются число и интервал повторения сигнала. В ходе последовательного нажатия выбранной функциональной клавиши на клавиатуре с использованием данной формы осуществляется само измерение. Фиксация результатов измерения обеспечивается нажатием кнопки Результат. В меню Результаты тестирования отображаются результаты проведенного измерения. Результаты можно сохранить в текстовом формате или/и в БД. Форма Информация о программе содержит в себе инструкции по работе с программным комплексом.

Расчет среднего прогнозируемого времени аудио-моторной реакции оператора можно провести на основе определения антилогарифма, получаемого по модели авторегрессии третьего порядка от логарифмов средних результатов измерений в одной серии наблюдений. Для определения по-

рядка авторегрессии использовался подход, предложенный в [9]. В качестве основы для разработки математического обеспечения прогнозирования интегральной характеристики функционального состояния нервной системы по данным времени аудио-моторной реакции может использоваться методика из [10]. Время аудио-моторной реакции является одной из составляющих для оценки психомоторики оператора и выбора стратегии управления химико-технологическим процессом на основе минимизации потерь в ходе человеко-машинного взаимодействия.

Разработанный программный комплекс может выступить в качестве прототипа серийной разработки ПО для получения и прогнозирования оценок состояния центральной нервной системы оператора при функционировании в составе АСУ.

Литература

1. Епанешникова Е.К., Федоров В.К. Проблемы изучения человеческого фактора в инновационных процессах // Инновации. 2008. № 3. С. 56–59.
2. Нехорошкова А.Н., Грибанов А.В., Депутат И.С. Сенсомоторные реакции в психофизиологических исследованиях (обзор) // Вестн. САФУ. Сер. Медико-биологические науки. 2015. № 1. С. 38–48.
3. Шутова С.В., Муравьева И.В. Сенсомоторные реакции как характеристика функционального состояния ЦНС // Вестн. ТвГУ. 2013. Т. 18. Вып. 5. С. 2831–2840.
4. Ахремчик О.Л. Характеристики аппаратно-программных комплексов для психофизиологического тестирования обучаемых // Вестн. ТвГУ. 2014. № 2. С. 38–41.
5. Слета В., Гурьянов Л. От измерения и обработки тегов к объектам и быстрой разработке автоматизированных систем // Control engineering Россия. 2015. № 6. С. 20–23.
6. Дюк В.А., Эммануэль В.Л. Информационные технологии в медико-биологических исследованиях. СПб: Питер, 2003. 528 с.
7. Кулаичев А.П. Компьютерная электрофизиология в клинической и исследовательской практике. М.: 1999. 330 с.
8. Бобнев М.П. Генерирование случайных сигналов. М.: Энергия, 1971. 239 с.
9. Akaike H. Fitting autoregressive models for prediction. Annals of the institute of statistical mathematics, 1969, no. 21, pp. 243–247.
10. Лоскутова Т.Д. Оценка функционального состояния центральной нервной системы человека по параметрам простой двигательной реакции // Физиологический журнал СССР им. И.М. Сеченова. 1975. № 1. С. 3–11.

A SOFTWARE PACKAGE TO MEASURE TIME OF AUDIO MOTOR REACTION OF CHEMICAL TECHNOLOGICAL PROCESS CONTROL SYSTEM OPERATORS

*O.L. Akhremchik*¹, Dr.Sc. (Engineering), Associate Professor;
*I.I. Bazulev*¹, Postgraduate Student, bazulevivan@gmail.com

¹ Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation

Abstract. Time of the operator's sensory-motor response is a basis for choosing models of information representation and the strategy of chemical and technological process control to minimize losses during human-computer interaction. Sound signal application for forming messages of abnormal and preventive alarms leads to the need of development of methods and monitoring aids of audio-motor response time.

The paper considers the structure, functional characteristics and the main components of a software package for measuring simple audio motor reaction time of a control system operator. According to the structural programming conception, Delphi means are selected to implement the complex. The paper describes screen forms types of the program complex and possible operator actions to use the forms.

As complex set-up parameters it is offered to use a type of a sound signal; time interval defining the moment of signal appearance; a number of signal presentations. The different frequency tone signals are used as incentives for measurement. Databases of sound signals, operators and measurement results are filled and used during measurement.

The software package might be applied to operate as a part of a control system of chemical and technological process. The package may be considered as a tool for indirect estimation of the status of the operator's central nervous system. A basis of measurement results transfer to other program systems is saving results as *.cds and *.txt.

The package approbation results show that the nature of influence of external factor set on operator's audio motor reaction time has a multiplicative character.

Keywords: audio signal, time, operator, software package, reaction.

References

1. Epaneshnikova E.K., Fedorov V.K. The problems of studying a human factor in innovation processes. *Innovatsii* [Innovations]. 2008, no. 3 (113), pp. 56–59 (in Russ.).
2. Nekhoroshkova A.N., Gribanov A.V., Deputat I.S. Sensorimotor reactions in psychophysiological studies (review). *Vestnik Severnogo (Arkticheskogo) federalnogo universiteta. Ser. "Mediko-biologicheskie nauki"* [Vestnik of Northern (Arctic) Federal Univ. Jour. of Medical and Biological Research]. 2015, no. 1, pp. 38–48 (in Russ.).
3. Shutova S.V., Muravyeva I.V. Sensorimotor reactions as characteristics of functional state of CNS. *Vestnik TGU. Ser. Estestvennye i tekhnicheskie nauki* [Tambov University Reports. Series Natural and Technical Sciences]. 2013, vol. 18, iss. 5, pp. 2831–2840 (in Russ.).
4. Akhremchik O.L. Characteristics of hardware and software packages for psychophysiological testing of trainees. *Vestnik Tverskogo gos. tekhnicheskogo univ.* [Bulletin of Tver State Technical Univ.]. 2014, no. 2 (26), pp. 38–41 (in Russ.).
5. Sleta V., Guryanov L. From measuring and tag processing to objects and fast development of automated systems. *Control engineering Rossiya* [Control Engineering Russia]. 2015, no. 6, pp. 20–23 (in Russ.).
6. Dyuk V.A., Emmanuel V.L. *Informatsionnye tekhnologii v mediko-biologicheskikh issledovaniyakh* [Information Technologies in Biomedical Research]. St. Petersburg, Piter Publ., 2003, 528 p.
7. Kulaichev A.P. *Kompyuternaya elektrofiziologiya v klinicheskoy i issledovatel'skoy praktike* [Computer Electrophysiology in Clinical and Research Practice]. Moscow, 1999, 330 p.
8. Bobnev M.P. *Generirovanie sluchaynykh signalov* [Random Signal Generation]. Moscow, Energiya Publ., 1971, 239 p.
9. Akaike H. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*. 1969, no. 21, pp. 243–247.
10. Loskutova T.D. Estimation of human central nervous system functional status by simple motor reaction parameters. *Fiziologicheskyy zhurnal SSSR im. I.M. Sechenova* [Physiological Journal n.a. I.M. Sechenov]. 1975, no. 1, pp. 3–11 (in Russ.).

Примеры библиографического описания статьи

1. Ахремчик О.Л., Базулев И.И. Программный комплекс для измерения времени аудио-моторных реакций операторов систем управления химико-технологическими процессами // Программные продукты и системы. 2017. Т. 30. № 2. С. 328–332; DOI: 10.15827/0236-235X.118.328-332.
2. Akhremchik O.L., Bazulev I.I. A software package to measure time of audio motor reaction of chemical technological process control system operators. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 2, pp. 328–332 (in Russ.); DOI: 10.15827/0236-235X.118.328-332.

УДК 621.311
DOI: 10.15827/0236-235X.118.333-337

Дата подачи статьи: 12.01.17
2017. Т. 30. № 2. С. 333–337

О ПРОГРАММЕ МОДЕЛИРОВАНИЯ УЗЛОВЫХ ЦЕН НА ЭЛЕКТРОЭНЕРГИЮ

*Э.Д. Халилов, к.т.н., доцент, начальник отдела, elmanxalilov2010@mail.ru
(Азербайджанский научно-исследовательский и проектно-изыскательский институт
энергетики, ул. Зардаби Гасанбека, 94, г. Баку, Аз1012, Азербайджан)*

В условиях реформ системы электроснабжения разных стран, в том числе Азербайджана, появилась возможность использования возобновляемых источников электроэнергии в электрической сети. Актуальными стали исследования, основанные на прослеживании потоков мощности. Появились исследования прослеживания мощности «зеленых» генераторов и мощности искажений источников гармоник.

Торговля электроэнергией на оптовом рынке осуществляется с использованием таких механизмов, как рынок на сутки вперед, балансирующий рынок, рынок регулируемых договоров, свободные договора купли-продажи электроэнергии.

Различная стоимость электроэнергии у разных поставщиков обусловила актуальность задачи определения узловых цен. Узловые цены определяют стоимость единицы электроэнергии, покупаемой в узле, и включают в себя маргинальную цену балансирующего узла, потери в линии электропередачи, цену ограничений линии электропередачи. Для расчета этих составляющих требуется соответствующее ПО.

Разработан алгоритм для моделирования узловых цен на рынке электроэнергии. Он реализован в виде программы в среде MATLAB. Работоспособность программы проверена на различных тестовых схемах.

Приведены результаты расчетов узловых цен на примере семиузловой тестовой схемы электрической сети.

Ключевые слова: рынок электроэнергии, узловые цены, моделирование, алгоритм, активная мощность, фиктивные узлы, фиктивные ветви, MATLAB.

В условиях продолжающихся реформ отрасли электроснабжения появилась возможность для широкого включения возобновляемых источников электроэнергии (ЭЭ) в электрической сети. Стали актуальными исследования, основанные на прослеживании потоков мощности. В последние годы появляются исследования по отслеживанию мощности «зеленых» генераторов и мощности искажений источников гармоник. Для решения задачи адресности мощностей нашли применение матричный подход, развитый Биалеком, и подход, основанный на применении теории графов [1–3].

Как товар ЭЭ является предметом купли-продажи. Торговля ЭЭ на оптовом рынке осуществляется с использованием таких рыночных механизмов, как рынок на сутки вперед, балансирующий рынок, рынок регулируемых договоров, свободные договоры купли-продажи [4, 5].

Рынок на сутки вперед – это аукцион ценовых заявок на продажу и покупку ЭЭ, проводимый на каждый час за сутки до фактической поставки ЭЭ. В рамках торговли на рынке на сутки вперед поставщики и покупатели подают ценовые заявки коммерческому оператору, в которых указывают, какой объем ЭЭ в какой час и по какой цене они готовы поставить или купить соответственно.

Конкурентному рынку присуще заблаговременное планирование режимов работы генерирующего оборудования производителей ЭЭ, что связано с невозможностью одновременного ввода в работу этого оборудования. Возможность планирования на рынке на сутки вперед соответствует времени пуска генерирующего оборудования.

Балансирующий рынок – рынок отклонений в реальном времени. На балансирующем рынке про-

исходит купля-продажа объемов ЭЭ, потребленных сверх плановых объемов или не потребленных относительно плановых объемов.

Особенностью балансирующего рынка ЭЭ является то, что на нем не определяется состав оборудования. Основной задачей этого рынка является оптимальное экономическое и технологическое поддержание баланса ЭЭ.

На энергетических рынках мира наибольшее распространение получили две основные модели: пул и двусторонние контракты. Пул чаще используется на оперативных рынках и рынках на сутки вперед, а двойной контракт – при долгосрочных контрактах (от недели до нескольких лет) [4, 5].

Оптимизационная модель рынка ЭЭ

Целевая функция (социальное благосостояние) рынка ЭЭ имеет вид

– в одностороннем аукционе:

$$\min \sum_{j=1}^n c_{g_j} P_{g_j} ; \max \sum_{i=1}^m c_{d_i} P_{d_i} ; \quad (1)$$

– в двухстороннем аукционе:

$$\max \left\{ \sum_{i=1}^m c_{d_i} P_{d_i} - \sum_{j=1}^n c_{g_j} P_{g_j} \right\}. \quad (2)$$

При этом учитываются балансовые ограничения активной и реактивной мощности в узлах, сетевые ограничения, то есть ограничения перетоков активной мощности по контролируемым сечениям, а также ограничения на диапазон изменения оптимизируемых переменных [6–10]:

$$\sum_i P_{ij} + \sum_g P_g - \sum_c P_c = 0, \quad \sum_i Q_{ij} + \sum_g Q_g - \sum_c Q_c = 0,$$

$P_s^{\min} \leq \sum_{ij \in s} P_{ij} \leq P_s^{\max}$, $0 \leq P_g \leq P_g^{\max}$, $0 \leq Q_g \leq Q_g^{\max}$,
 $0 \leq P_c \leq P_c^{\max}$, $U_j^{\min} \leq U_j \leq U_j^{\max}$, где d – потребление в узле; g – генерация; c_{d_i} – узловая цена покупки активной мощности; c_{g_j} – узловая цена продажи активной мощности; P_{di} – узловое потребление активной мощности; P_{gi} – узловая генерация активной мощности; n , m – количество заявок на производство и потребление активной мощности; U_j – узловое напряжение; P_{ij} и Q_{ij} – перетоки мощности из узла j в узел i по ветви ij ; s – контролируемые сечения.

В односторонних аукционах присутствуют отдельные составляющие целевой функции и целевая функция представлена компонентами, соответствующими ценовым заявкам продавцов или покупателей.

Коммерческие отношения в *электроэнергетических системах* (ЭЭС) приводят к необходимости использования финансово-технологических моделей систем для успешного функционирования электроэнергетической отрасли. В этих моделях рассматриваются не только технологические переменные, описывающие режим ЭЭС, связывающие эти переменные ограничения в виде равенств и неравенств, но и переменные, учитывающие финансовые факторы и уравнения финансовых балансов, связывающие эти переменные. Финансово-технологические модели ЭЭС позволяют имитировать поведение ЭЭС в различных финансовых и технологических ситуациях. Данные модели используют информацию об адресности генерации перетоков мощности и нагрузок [6, 7].

Математические модели узловых цен

Важнейшими финансовыми показателями являются узловые цены, они отражают стоимость единицы ЭЭ, покупаемой в узле. Среди посвященных узловым ценам работ следует отметить [6–10].

Узловые цены определяют стоимость единицы ЭЭ, покупаемой в узле, и включают в себя маргинальную цену балансирующего узла, потери в линии электропередачи, цену ограничений линии электропередачи. Расчет этих составляющих требует разработки соответствующего ПО.

В [6] для получения узловых цен использованы финансово-технологические модели. В [7] узловые средневзвешенные цены для взаиморасчетов потребителей и производителей ЭЭ рассчитываются на основе алгоритмов определения средневзвешенных цен по зафиксированным потокам ЭЭ и ценовым заявкам источников. Анализируется чувствительность узловых цен к ценовым заявкам и изменениям перетоков.

В [8] для моделирования рынка ЭЭ применен механизм аукциона. После отбора наиболее эконо-

мичных генерирующих мощностей при соблюдении всех ограничений формируется двойственная задача линейного программирования. В результате решения этой задачи определяются стоимость достигнутых ограничений по связям и узловые цены. На основании этих данных и с учетом правил принятия решения можно проанализировать эффективность подачи участником ценовых заявок.

В [9] предложена система интегральных показателей разницы узловых цен, включающая стоимостные, ценовые и физические показатели. Предложенная система позволяет количественно оценить влияние режимов ЭЭС на разницу цен в узлах при решении оптимизационной задачи оптового рынка ЭЭ.

В [10] на примере тестовой девятиузловой схемы ЭЭС рассматривается влияние изменения ценовых заявок производителей ЭЭ на узловые цены для потребителей. Производятся анализ полученных результатов и их обобщение. В данной работе узловые цены определяются совместной реализацией методов линейного и нелинейного программирования с применением метода неопределенных множителей Лагранжа.

В [11] разработана математическая модель, в которой режимные ограничения учитываются методом Лагранжа, а потери – с использованием R-схемы и коэффициентов токораспределения.

Математические модели узловых цен могут быть использованы в качестве информационной поддержки для системного оператора, администратора торговой системы. Участники рынка, используя узловые цены, полученные с помощью этих моделей, могут правильно выбрать стратегию поведения на рынке. Кроме этого, узловые цены обеспечивают адекватное распределение платы за ЭЭ. При этом потребитель платит пропорционально затратам на выработку потребленной им ЭЭ.

В данной статье рассматриваются матричный алгоритм и программа для моделирования узловых цен с возможностью стыковки с программой токораспределения или оценивания состояния.

Моделирование и результаты расчетов узловых цен

Основу матричного метода составляет уравнение

$$G_i(P_{gi}) + \sum_j c_j P_{ji} = c_i \cdot P_{Hi} + c_i \cdot \sum_j P_{ji}, \quad (3)$$

$i = 1, \dots, n$.

В данном уравнении первое слагаемое в левой части – затраты на производство мощности P_{gi} , второе слагаемое – затраты на покупку перетоков мощности P_{ji} в i -й узел из узлов j по цене c_j [6, 7]. Первое слагаемое в правой части – выручка от продажи нагрузки P_{Hi} в узле i , второе – выручка от продажи перетоков мощности P_{ij} из узла i по цене c_i .

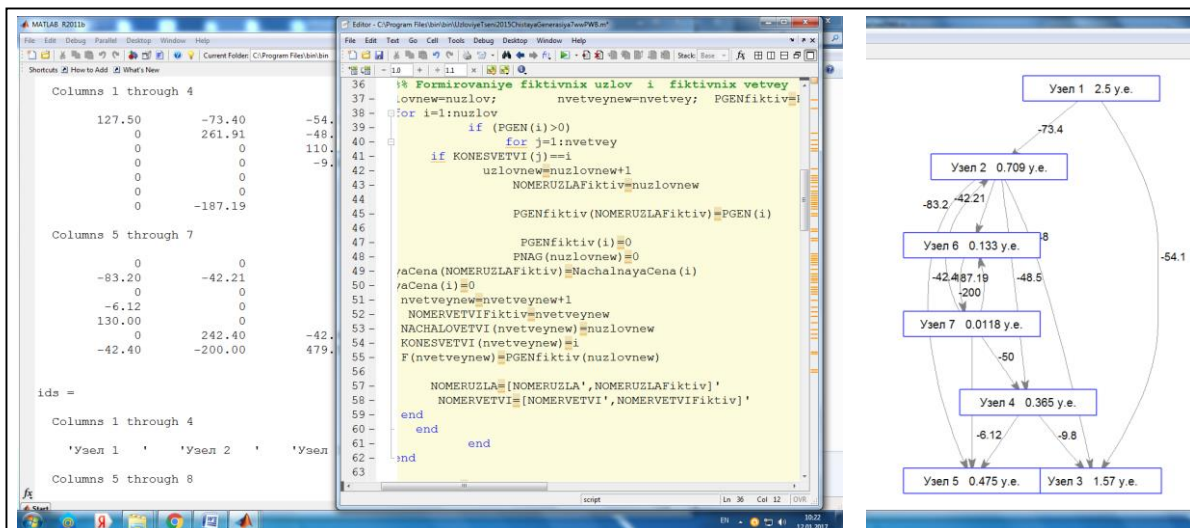


Рис. 1. Программа моделирования узловых цен

Fig. 1. The program for nodal prices modeling

Следует отметить, что для учета потерь ΔP_{ij} в ветви $i-j$ цена c_j увеличивается до значения c_{ij} , определяемой из условия $c_{ij} = \frac{c_j P_{ji}}{P_{ij}}$, где $P_{ji} = P_{ij} + \Delta P_{ij}$.

В матричной форме уравнение (3) можно представить в виде

$$A \cdot c = G, \quad (4)$$

где G – ценовая заявка генераторов или их расходные характеристики; c – узловые цены. Диагональные элементы матрицы A представляют собой сумму нагрузки в узле с отходящими потоками:

$$A_{ii} = P_{n,i} + \sum_{j=1}^K P_{ij}, \quad (5)$$

где K – число узлов, в которые текут потоки мощности из узла i . Недиagonalными элементами матрицы A являются втекающие в узел потоки мощности:

$$A_{ij} = -P_{ji}. \quad (6)$$

Разработан алгоритм для моделирования узловых цен, состоящий из шести этапов.

1. Расчет установившегося режима сети.
2. Передача необходимой информации об узлах и ветвях в программу узловых цен.
3. Формирование матриц узловых инъекций мощности, потоков мощности.
4. Формирование фиктивных узлов и фиктивных ветвей.
5. Формирование матрицы потоков мощности.
6. Определение узловых цен.

Разработанный алгоритм реализован в виде MATLAB-программы (рис. 1). Данные по узлам и ветвям, а также результаты расчета установившегося режима схемы приведены в таблицах 1 и 2 соответственно.

Таблица 1
Исходные данные семиузловой схемы (по узлам)

Table 1

Initial data for 7-node scheme (node data)

Узел	Модуль и угол напряжения		Нагрузка		Генерация	
	U (о.е.)	δ (град.)	P_n (МВт)	Q_n (МВАр)	P_r (МВт)	Q_r (МВАр)
1	1.05	6.4			127.54	18.02
2	1.04	4.13	40	20	187.19	46.02
3	0.99	-0.34	110	40		
4	1	-0.2	80	30	50	14.57
5	1.01	-1.17	130	40		
6	1.04	2.75	200	0	200.17	-6.58
7	1.04	0	200	0	200.49	38.64

Таблица 2
Исходные данные семиузловой схемы (по ветвям)

Table 2

Initial data for 7-node scheme (branch data)

Номер ветви	От узла	К узлу	P_{ij} , МВт
1	1	2	73.4
2	1	3	54.1
3	2	3	48
4	2	4	46.5
5	2	5	83.2
6	2	6	42.2
7	3	4	-9.8
8	4	5	6
9	7	5	42.4
10	6	7	42.4

Работоспособность программы проверена на различных тестовых схемах IEEE. На рисунке 2 изображена семиузловая тестовая схема [12]. В таблице 3 приведены результаты расчета узловых цен.

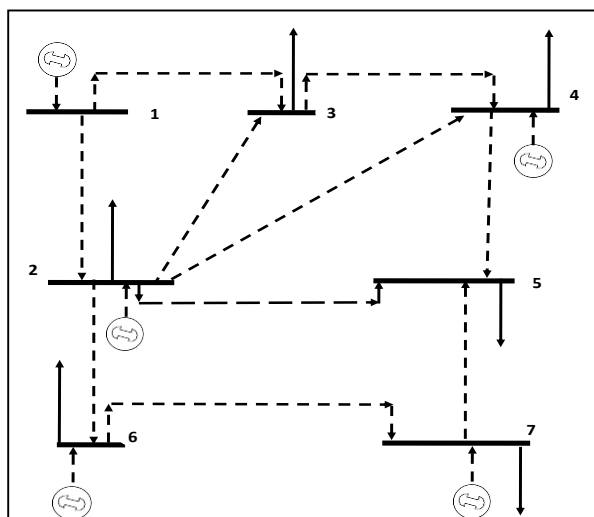


Рис. 2. Тестовая семиузловая схема

Fig. 2. 7-node test scheme

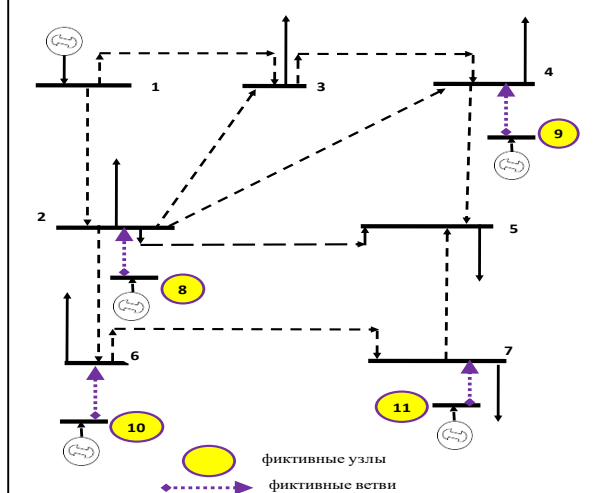


Рис. 3. Тестовая схема после добавления фиктивных узлов и ветвей

Fig. 3. The test scheme after adding fictitious nodes and branches

Таблица 3

Изменение узловых цен в ходе расчета

Table 3

Nodal price changes during calculation

Но- мер узла	Рген0, МВт	Рген, МВт	Исходная уз- ловая цена (у.е.)	Узловая цена после расчета (у.е.)
1	127.54	127.54	2.5	2.5
2	187.19	0	3	2.85
3	0	0	0	2.79
4	50	0	4	3.52
5	0	0	0	2.59
6	200.17	0	2	2.15
7	200.49	0	1.8	1.86
8	-	187.19	-	3
9	-	50	-	4
10	-	200	-	2
11	-	200	-	1.8

В ходе работы программа автоматически формирует списки фиктивных узлов и фиктивных ветвей для получения чистой генерации. На рисунке 3 приведена тестовая схема после добавления фиктивных узлов и ветвей. Как видим, к узлам 2, 4, 6 и 7 добавляются фиктивные ветви. В результате такого преобразования генерация в узлах 2, 4, 6 и 7 присвоена фиктивным узлам 8, 9, 10 и 11. В таблице 4 дана информация по ветвям после окончания расчетов. Выделенная часть в таблицах 2 и 3 соответствует фиктивным узлам и ветвям соответственно.

Таблица 4

Информация по ветвям после окончания расчетов

Table 4

The information on branches after calculations

Номер ветви	От узла	К узлу	P_{ij}
1	1	2	73.4
2	1	3	54.1
3	2	3	48
4	2	4	48.5
5	2	5	83.2
6	2	6	42.2
7	3	4	-9.8
8	4	5	6
9	7	5	42.4
10	6	7	42.4
11	8	2	187.19
12	9	4	50
13	10	6	200
14	11	7	200

Результаты расчетов узловых цен показывают, что в разных узлах ЭЭС образуются разные узловые цены, которые могут быть использованы в рыночных условиях при ценовых заявках.

Выводы

В статье описаны разработанные алгоритм и программа в среде MATLAB для моделирования узловых цен на основе матричного подхода. На примере семиузловой схемы электрической сети проведены расчеты узловых цен. Результаты расчетов по разработанной программе могут быть использованы в рыночных условиях при ценовых заявках.

Литература

1. Bialek J.W. Tracing the flow of electricity. IEE Proc-Gener., Transm., and Distrib., Jul. 1996, vol. 143, pp. 310–320.
2. Гамм А.З., Голуб И.И. Адресность передачи активных и реактивных мощностей в электроэнергетической системе // Электричество. 2003. № 3. С. 9–16.
3. Баламетов А.Б., Халилов Э.Д. О прослеживании путей передачи мощностей в энергосистемах // Проблемы энергетики. 2011. № 4. С. 10–17.
4. Стофт С. Экономика энергосистем. Введение в проектирование рынков электроэнергии; [пер. с англ.]. М.: Мир, 2006. 623 с.
5. Беляев Л.С. Проблемы электроэнергетического рынка. Новосибирск: Наука, 2009. 296 с.
6. Гамм А.З., Войтов О.Н., Голуб И.И., Бровяков Ю.А. Технико-экономические модели электроэнергетических систем // Информационные технологии контроля и управления на

транспорте: сб. науч. тр. Иркутск: Изд-во ИРИИТ, 2000. Вып. 7. С. 145–154.

7. Гамм А.З., Голуб И.И., Батюнин А.В., Гамм А.А. Узловые средневзвешенные цены на электроэнергию // *Электричество*. 2005. № 10. С. 17–24.

8. Аюев Б.И., Ерохин П.М., Паниковская Т.Ю. Применение механизма аукциона для моделирования рынка электроэнергии // *Изв. Томского политехнич. ун-та*. 2005. Т. 308. № 5. С. 176–178.

9. Васильковская Т.А. Показатели разницы узловых цен на

оптовом рынке электроэнергии // *Электричество*. 2007. № 2. С. 23–27.

10. Mahnitko A., Gerhards J., Ribakovs S., Umbraško I. Pricing questions in the electric power markets. *Power and Electrical Engineering*, 2009, part 4, vol. 24, pp. 16–23.

11. Булатов Б.Г., Каркунов В.О. Упрощенная модель определения узловых цен на рынке электроэнергии // *Вестн. ЮУрГУ. Сер. Энергетика*. 2009. Вып. 12. № 34. С. 11–14.

12. PowerWorld. URL: <http://www.powerworld.com/products/simulator/overview> (дата обращения: 11.01.2017).

Software & Systems

DOI: 10.15827/0236-235X.118.333-337

Received 12.01.17

2017, vol. 30, no. 2, pp. 333–337

ON MODELING OF ELECTRICITY NODAL PRICES

E.D. Khalilov¹, Ph.D. (Engineering), Associate Professor, Head of Department, elmanxalilov2010@mail.ru

¹ *Azerbaijan Scientific-Research and Design-Prospecting Power Engineering Institute, Zardabi St. 94, Baku, AZ1012, Azerbaijan*

Abstract. In terms of power supply system reforms in various countries, including Azerbaijan, there is a possibility of using renewable energy sources in the electrical network. Relevant studies began to trace the power of “green” power generators and harmonic distortion sources.

Electrical energy trading on the wholesale market uses market mechanisms, such as a day-ahead market, a balancing market, a market of regulated contracts, and free electricity sale and purchase agreements.

As generating stations and other suppliers have variable cost of electricity, this fact leads to solving the relevant problem of determining nodal prices. Nodal prices show a electricity unit cost purchased in a node. Each node price includes such components as a marginal price of a balancing node, losses in transmission lines, power line price restrictions. Calculation of these components requires development of the appropriate software.

The paper describes the developed algorithm for simulation of nodal prices. This algorithm is implemented as a program in MATLAB system to determine nodal prices in the electricity market. The software efficiency is tested at different test schemes. In particular, this article presents the results of nodal price calculations using the example of 7-node test scheme of electrical network.

Keywords: electricity market, nodal prices, modeling, algorithm, active power, fictitious nodes, fictitious branches, MATLAB.

References

1. Bialek J.W. Tracing the Flow of Electricity. *IEE Proc-Gener., Transm., and Distrib.* 1996, vol. 143, pp. 310–320.
2. Gamm A.Z., Golub I.I. Address capacity of power transfer in an electrical power system. *Elektrichestvo* [Electricity]. 2003, no. 3, pp. 9–16 (in Russ.).
3. Balametov A.B., Halilov E.D. On tracking transmission routes in electric systems. *Problemy energetiki* [Power Industry Problems]. Baku, 2011, no. 4, pp. 10–17 (in Russ.).
4. Stoff S. *Power System Economics: Designing Markets for Electricity*. Wiley and Sons Publ., 2002, part 1.
5. Belyaev L.S. *Problemy elektroenergeticheskogo rynka* [Electrical Power Market Problems]. Novosibirsk, Nauka Publ., 2009, 296 p.
6. Gamm A.Z., Voytov O.N., Golub I.I., Brovyakov Yu.A. Technical and economic models of electrical power systems. *Informatsionnye tekhnologii kontrolya i upravleniya na transporte* [Information Technologies of Transport Control and Management]. Irkutsk, IrIIT Publ., 2000, iss. 7, pp. 145–154 (in Russ.).
7. Gamm A.Z., Golub I.I., Batyunin A.V., Gamm A.A. Node weighted average electricity prices. *Elektrichestvo* [Electricity]. 2005, no. 10, pp. 17–24 (in Russ.).
8. Ayuev B.I., Erokhin P.M., Panikovskaya T.Yu. Using the auction mechanism for modeling an electrical power market. *Izvestiya Tomskogo politehnicheskogo universiteta* [Bulletin of the Tomsk Polytechnic Univ.]. 2005, vol. 308, no. 5, pp. 176–178 (in Russ.).
9. Vaskovskaya T.A. Level of node price margin in the electrical power market. *Elektrichestvo* [Electricity]. 2007, no. 2, pp. 23–27 (in Russ.).
10. Mahnitko A., Gerhards J., Ribakovs S., Umbraško I. Pricing questions in the electric power markets. *Power and Electrical Engineering*. Riga, RTU Publ., 2009, part 4, vol. 24, pp. 16–23.
11. Bulatov B.G., Karkunov V.O. Simplified methods of nodal prices determination at the electric energy market. *Vestnik YuUrGU* [Bulletin of the South Ural State Univ.]. Series Power Engineering, 2009, no. 34, pp. 11–14 (in Russ.).
12. *PowerWorld Simulator*. Available at: <http://www.powerworld.com/products/simulator/overview> (accessed January 11, 2017).

СОДЕРЖАНИЕ

Ляковский В.Л., Бреслер И.Б., Алашеев М.А.

Постановка задачи формирования направлений развития автоматизированных систем
организационного типа и алгоритм ее решения 165

Еделев А.В., Зоркальцев В.И., Доан Ван Бинь, Нгуен Хоай Нам

Формирование вариантов развития энергетики Вьетнама
методами комбинаторного моделирования 172

Шибзухов З.М., Димитриченко Д.П., Казаков М.А.

Принцип минимизации эмпирического риска на основе агрегирующих функций
средних потерь для решения задач регрессии 180

Быкова В.В., Катаева А.В.

Сжатое представление строгих ассоциативных правил в анализе данных 187

Billig V.A.

Effective algorithm for constructing associative rules 196

Фомин В.В., Александров И.В.

Параллельные вычисления при реализации web-инструментария распознавания образов
на основе методов прецедентов 207

Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.

Однородная распределительная задача на основе моделей адаптивного поведения
муравьиной колонии 217

Пальчевский Е.В., Халиков А.Р.

Автоматизированная система обработки данных в UNIX-подобных системах 227

Афанасьев А.П., Дзюба С.М., Емельянова И.И.

Обобщенно-периодические движения динамических и неавтономных
периодических систем 235

Есиков Д.О. Оценка эффективности методов решения задач обеспечения

устойчивости функционирования распределенных информационных систем 241

Агамиров Л.В., Вестяк В.А., Агамиров В.Л.

Рекурсивный алгоритм точного расчета ранговых критериев проверки
статистических гипотез 257

Григораш А.С., Курейчик В.М., Курейчик В.В.

Программный комплекс решения задачи кластеризации 261

Свириденко А.Б.

Реализация и тестирование ньютоновских методов безусловной оптимизации 270

Бакулина М.П.

Эффективный метод блочного кодирования двухуровневых изображений 282

Кудряшов А.П., Соловьев И.В.

Текстурирование трехмерной модели, построенной по объединенному набору видов с зашивкой дыр 286

Чижикова Л.А.

Принципы проектирования модульной архитектуры программного обеспечения авиационной тематики 291

Абу-Абед Ф.Н., Допира Р.В., Ищук И.Н., Брежнев Е.Ю., Потапов А.Н.

Унифицированное описание функционирования информационных радиоэлектронных систем для оценки программного обеспечения учебно-тренировочных средств 301

Матвейкин В.Г., Дмитриевский Б.С., Медников В.И., Семержинский С.Г.

Программно-алгоритмический комплекс защиты и управления предприятием 307

Дроботун Е.Б., Угловский Е.П., Замалтдинов И.Ш.

Метод формирования множеств альтернативных вариантов построения подсистем, входящих в состав системы защиты от компьютерных атак 314

Голубчиков С.В., Новиков В.К., Баранова А.В.

Уровни и правовая модель информационной безопасности (защиты информации) 320

Маркина Т.А., Хрупов В.А.

Исследование защищенности систем электронных платежей 324

Ахремчик О.Л., Базулев И.И.

Программный комплекс для измерения времени аудио-моторных реакций операторов систем управления химико-технологическими процессами 328

Халилов Э.Д.

О программе моделирования узловых цен на электроэнергию 333

C O N T E N T

Lyaskovsky V.L., Bresler I.B., Alashev M.A.

Statement of the problem of formation of directions of development
of organizational automated systems and its solution algorithm..... 165

Edelev A.V., Zorkaltsev V.I., Đoàn Văn Bình, Nguyễn Hoài Nam

Formation of Vietnam energy development options
by combinatorial modeling methods 172

Shibzukhov Z.M., Dimitrichenko D.P., Kazakov M.A.

The empirical risk minimization principle based on average loss aggregating functions
for regression problems 180

Bykova V.V., Kataeva A.V.

A contracted representation of strong associative rules in data analysis 187

Billig V.A.

Effective algorithm for constructing associative rules 196

Fomin V.V., Aleksandrov I.V.

Parallel computing when implementing web-based pattern recognition tools
based on use case methods 207

Lebedev B.K., Lebedev O.B., Lebedeva E.M.

A homogeneous distribution problem based on ant colony adaptive behavior models 217

Palchevsky E.V., Khalikov A.R.

Automated data processing system in UNIX-like systems..... 227

Afanasiev A.P., Dzyuba S.M., Emelyanova I.I.

Generalized periodic motions of dynamical and non-autonomous periodic systems 235

Esikov D.O.

Evaluating the effectiveness of sustainability problem solving methods
of distributed information system functioning..... 241

Agamirov L.V., Vestyak V.A., Agamirov V.L.

Recursive algorithm for exact calculation of rank tests
for testing statistical hypotheses 257

Grigorash A.S., Kureychik V.M., Kureychik V.V.

Clustering software solution..... 261

Sviridenko A.B. Realization and testing of newton methods for unconstrained optimization..... 270

Bakulina M.P. An efficient block coding method for binary images 282

Kudryashov A.P., Solovov I.V. Texturing of 3d models with filled holes 286

Chizhikova L.A. Design principles of modular software architecture in aviation 291

Abu-Abed F.N., Dopira R.V., Ishchuk I.N., Brezhnev E.Yu., Potapov A.N.

Unified description of information radioelectronic system functioning
to estimate training aids software 301

Matveykin V.G., Dmitrievsky B.S., Mednikov V.I., Semerzhinsky S.G.

A software and algorithm complex of enterprise protection and management 307

Drobotun E.B., Uglovsky E.P., Zamaltdinov I.Sh.

Method of generating sets of alternative variants of building subsystems
which are a part of a computer attack protection system..... 314

Golubchikov S.V., Novikov V.K., Baranova A.V.

A system and legal model of information security (data protection) 320

Markina T.A., Khrupov V.A. Investigation of security of electronic payment systems 324

Akhremchik O.L., Bazulev I.I. A software package to measure time of audio motor reaction

of chemical technological process control system operators 328

Khalilov E.D. On modeling of electricity nodal prices 333

Вниманию подписчиков

Международный журнал
«Программные продукты и системы»
распространяется по подписке
в странах СНГ и за их пределами.
В Каталоге подписных изданий
Агентства «Роспечать» индекс журнала
70799

Телефон (482-2) 39-91-49

Факс (482-2) 39-91-00

www.swsys.ru
red@cps.tver.ru