

ISSN 2305-9052 (Print)
ISSN 2410-7034 (Online)

ВЕСТНИК



ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

BULLETIN

OF THE SOUTH URAL
STATE UNIVERSITY

СЕРИЯ

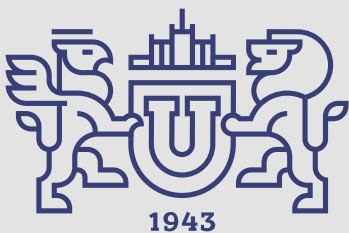
**ВЫЧИСЛИТЕЛЬНАЯ
МАТЕМАТИКА
И ИНФОРМАТИКА**

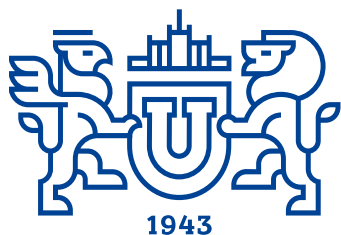
2021, том 10, № 2

SERIES

**COMPUTATIONAL
MATHEMATICS
AND SOFTWARE ENGINEERING**

2021, volume 10, no. 2





ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

2021
Т. 10, № 2

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий,
в которых должны быть опубликованы результаты диссертаций
на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет
(национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Искусственный интеллект и машинное обучение
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

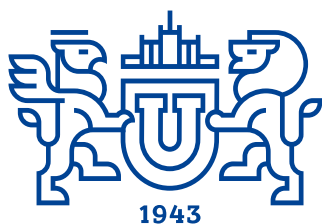
Редакционная коллегия

Л.Б. Соколинский, д.ф.-м.н., проф., *гл. редактор*
В.П. Танана, д.ф.-м.н., проф., *зам. гл. редактора*
М.Л. Цымблер, д.ф.-м.н., доц., *отв. секретарь*
Г.И. Радченко, к.ф.-м.н., доц.
Я.А. Краева, *техн. секретарь*

Редакционный совет

С.М. Абдуллаев, д.г.н., профессор
А. Андреяк, PhD, профессор (Германия)
В.И. Бердышев, д.ф.-м.н., акад. РАН, *председатель*
В.В. Воеводин, д.ф.-м.н., чл.-кор. РАН

Дж. Донгарра, PhD, профессор (США)
С.В. Зыкин, д.т.н., профессор
Д. Маллманн, PhD, профессор (Германия)
А.В. Панюков, д.ф.-м.н., профессор
Р. Продан, PhD, профессор (Австрия)
А.Н. Томилин, д.ф.-м.н., профессор
В.И. Ухоботов, д.ф.-м.н., профессор
В.Н. Ушаков, д.ф.-м.н., чл.-кор. РАН
М.Ю. Хачай, д.ф.-м.н., профессор
А. Черных, PhD, профессор (Мексика)
П. Шумяцкий, PhD, профессор (Бразилия)



BULLETIN

OF THE SOUTH URAL STATE UNIVERSITY 2021
vol. 10, no. 2

SERIES

“COMPUTATIONAL
MATHEMATICS AND SOFTWARE
ENGINEERING”

ISSN 2305-9052

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya “Vychislitel'naya Matematika i Informatika”

South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- Artificial intelligence and machine learning
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

Editorial Board

L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russia)
V.P. Tanana, South Ural State University (Chelyabinsk, Russia)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russia)
G.I. Radchenko, South Ural State University (Chelyabinsk, Russia)
Ya.A. Kraeva, South Ural State University (Chelyabinsk, Russia)

Editorial Council

S.M. Abdullaev, South Ural State University (Chelyabinsk, Russia)
A. Andrzejak, Heidelberg University (Germany)
V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
J. Dongarra, University of Tennessee (USA)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
D. Mallmann, Julich Supercomputing Centre (Germany)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russia)
R. Prodan, University of Innsbruck (Innsbruck, Austria)
P. Shumyatsky, University of Brasilia (Brazil)
A. Tchernykh, CICESE Research Center (Mexico)
A.N. Tomilin, Institute for System Programming of the RAS (Moscow, Russia)
V.I. Ukhobotov, Chelyabinsk State University (Chelyabinsk, Russia)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
V.V. Voevodin, Lomonosov Moscow State University (Moscow, Russia)
S.V. Zykin, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Omsk, Russia)

Содержание

ДИСКРЕТНЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ С ПОМЕХОЙ И ИХ ПРИЛОЖЕНИЯ К РЕШЕНИЮ ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ С.А. Никитина, В.И. Ухоботов	5
АВТОМАТИЧЕСКИЙ ПОДБОР ПАРАМЕТРОВ МОДЕЛИ ARIMA ДЛЯ ПРОГНОЗА КОЛИЧЕСТВА СЛУЧАЕВ ЗАРАЖЕНИЯ И СМЕРТИ ОТ COVID-19 Т.А. Макаровских, М.С.А. Аботалёб	20
О ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ Л.Б. Соколинский, И.М. Соколинская	38
ОСОБЕННОСТИ РЕАЛИЗАЦИИ АЛГОРИТМА TREECODE ДЛЯ РЕШЕНИЯ ЗАДАЧИ N-ТЕЛ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ А.В. Титов, А.В. Хоперсков	53
РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ИССЛЕДОВАНИЯ РЕСУРСА ПАРАЛЛЕЛИЗМА ЧИСЛЕННЫХ АЛГОРИТМОВ С ПОМОЩЬЮ ПРОГРАММНОЙ Q-СИСТЕМЫ В.Н. Алеева, П.С. Зотова, Д.С. Склезнев	66
КВАНТОВО-ХИМИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕРМОХИМИЧЕСКИХ СВОЙСТВ МОЛЕКУЛ ВЫСОКОЭНЕРГЕТИЧЕСКИХ ТЕТРАЗИНОВ В.М. Волохов, Е.С. Амосова, А.В. Волохов, Т.С. Зюбина, Д.Б. Лемперт, Л.С. Яновский, Д.А. Варламов	82

Contents

DISCRETE DYNAMIC SYSTEMS WITH INTERFERENCE AND THEIR APPLICATIONS TO THE SOLUTION OF THE PROBLEM OF RESERVE CONTROL S.A. Nikitina, V.I. Ukhobotov	5
AUTOMATIC SELECTION OF ARIMA MODEL PARAMETERS TO FORECAST COVID-19 INFECTION AND DEATH CASES T.A. Makarovskikh, M.S.A. Abotaleb	20
ON GENERATOR OF RANDOM PROBLEMS FOR LINEAR PROGRAMMING ON CLUSTER COMPUTING SYSTEMS L.B. Sokolinsky, I.M. Sokolinskaya	38
IMPLEMENTATION FEATURES OF THE TREECODE ALGORITHM FOR SOLVING N-BODY PROBLEMS ON GPU _s A.V. Titov, A.V. Khoperskov	53
ADVANCEMENT OF RESEARCH FOR THE PARALLELISM RESOURCE OF NUMERICAL ALGORITHMS WITH THE HELP OF SOFTWARE Q-SYSTEM V.N. Aleeva, P.S. Zotova, D.S. Skleznev	66
QUANTUM-CHEMICAL MODELING OF THERMOCHEMICAL PROPERTIES OF MOLECULES OF HIGH-ENERGY TETRAZINES V.M. Volokhov, E.S. Amosova, A.V. Volokhov, T.S. Zyubina, D.B. Lempert, L.S. Yanovskiy, D.A. Varlamov	82



This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

ДИСКРЕТНЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ С ПОМЕХОЙ И ИХ ПРИЛОЖЕНИЯ К РЕШЕНИЮ ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ

© 2021 С.А. Никитина, В.И. Ухоботов

Челябинский государственный университет
(454001 Челябинск, ул. Бр. Кашириных, д. 129)

E-mail: nikitina@csu.ru, ukh@csu.ru

Поступила в редакцию: 12.03.2021

Рассматриваются две модели дискретных динамических управляемых систем с помехой. В одной из них представлена дискретная задача управления, в которой вектограмма управления линейно зависит от заданных множеств. Во второй задаче предполагается, что вектограммы управления и помехи являются однотипными множествами. В обоих случаях цель выбора управления заключается в том, чтобы в момент окончания процесса управления фазовая точка содержалась в заданном множестве. При построении управления предполагается, что в каждый дискретный момент времени поступает информация о реализации помехи. Записан оператор программного поглощения, с помощью которого сформулированы условия на множество начальных положений, при которых гарантируется выполнение требуемого включения в заданный момент времени. В практической части работы показано применение полученных результатов на примере решения задачи управления запасами товара на складе. Пополнение товара происходит за счет его производства, а величина отгрузки товара определяется спросом. Предполагается, что о величине спроса на товар известно только множество его значений. Цель управления состоит в том, чтобы в заданный момент времени количество товара удовлетворяло определенным ограничениям. Получено множество начальных запасов товара, для которых возможно осуществить поставленную цель при любой реализации спроса.

Ключевые слова: дискретная система, задача управления запасами, вектограммы, линейно зависящие от заданных множеств, однотипная задача управления.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Никитина С.А., Ухоботов В.И. Дискретные динамические системы с помехой и их приложения к решению задачи управления запасами // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 5–19. DOI: 10.14529/cmse210201.

Введение

Создание и применение дискретных систем управления может быть вызвано несколькими причинами. Зачастую дискретные процессы управления возникают при решении практических задач. Это связано с тем, что информацию о состоянии управляемого процесса возможно получить только в дискретные моменты времени, а управление осуществляется по шагам. Кроме этого принцип действия некоторых элементов, входящих в систему, может быть дискретным. Отметим также, что для дискретных систем бывает проще реализовать сложные алгоритмы управления.

Задачи управления запасами составляют один из обширных классов дискретных задач управления, решение которых имеет важное прикладное значение. Определение стратегии управления запасами позволяет повысить эффективность используемых ресурсов.

В статье будут рассмотрены две модели дискретных динамических управляемых систем с помехой. Для каждой из них будет показано применение полученных теоретических результатов для решения задачи управления запасами.

Статья организована следующим образом. Раздел 1 содержит обзор работ по тематике исследования. В разделе 2 приводится решение задачи управления с помехой и вектограммами, линейно зависящими от заданных множеств. Раздел 3 посвящен рассмотрению дискретной задачи управления с помехой с однотипными вектограммами и выпуклой целью. В заключении приводится краткая сводка результатов, полученных в работе, и указаны направления дальнейших исследований.

1. Обзор работ

Для систем с дискретным временем в статье [14] решается задача адаптивного управления. В каждый момент времени управление выбирается на основе правила переключения. Задача управления состоит в том, чтобы определить подходящие правила для переключения и настройки параметров. Анализ динамических систем с дискретным временем, неопределенностями параметров и помехами представлен в работе [15]. Показано, что задача построения управления для таких систем сводится к решению линейных матричных неравенств. Алгоритм синтеза управления системой с дискретным временем при наличии внешнего воздействия был получен в [13], где также было показано применение полученной схемы на нескольких практических приложениях.

В работе [16] рассмотрена дискретная модель запасов с линейной скоростью спроса без регулярного предложения. Вводится критерий для расчета чистой прибыли, исходя из которого строится управление. В [17] представлены результаты моделирования динамики производства и заказов для системы поставок с дискретным временем. Получены оценки границ диапазона колебаний запасов при неизвестном спросе.

В работах [1–3] рассмотрена задача управления дискретной линейной системой с заданным моментом окончания и с фиксированным терминальным множеством. На каждом шаге при построении управления используется информация о реализации помехи. В статье [2] эта задача рассмотрена для случая, когда терминальное множество и вектограмма управления являются многогранниками специального вида, которые задаются с помощью системы линейных неравенств. Получены условия на множество начальных положений, при которых гарантируется достижение цели в момент окончания процесса управления. В [3] требуется, чтобы в момент окончания процесса управления фазовая точка содержалась в заданном множестве, имеющем форму кольца. Было построено семейство множеств, определяющее необходимые и достаточные условия возможности окончания. В работе [1] построены оптимальные управления для такой задачи.

В [4] предложен подход для решения задачи управления запасами в дискретном случае. Записан алгоритм построения управления, которое обеспечивает удержание фазовой точки в заданном семействе множеств при любой допустимой реализации помехи. В работе [12] была рассмотрена задача удержания фазовой точки в заданном семействе множеств в дискретные моменты времени, приведены необходимые и достаточные условия возможности удержания.

В данной статье продолжается начатое ранее исследование. Рассматриваются две модели дискретных динамических управляемых систем с помехой. В одной из них рассматривается дискретная задача управления с заданным моментом окончания, в которой вектограмма управления линейно зависит от заданных множеств, а в другой вектограммы управления и помехи являются однотипными множествами. Требуется, чтобы в момент окончания процесса управления фазовая точка содержалась в некотором множестве. В каждом из двух

случаев определены условия на множество начальных положений, при которых будет гарантироваться выполнение указанного требования.

2. Дискретная задача управления с помехой и вектограммами, линейно зависящими от заданных множеств

2.1. Постановка задачи

Пусть задан управляемый процесс

$$z(k+1) = z(k) - \sum_{i=1}^N a_i(k)u_i(k) + \sum_{j=1}^M b_j(k)v_j(k), \quad (1)$$

где $k = 0, \dots, p$, $z(k) \in \mathbb{R}^n$, $u_i(k) \in U_i$, $v_j(k) \in V_j$, $a_i(k) \geq 0$, $b_j(k) \geq 0$, p — фиксированный момент времени; $U_i \subset \mathbb{R}^n$ и $V_j \subset \mathbb{R}^n$ — выпуклые множества.

Задано выпуклое множество $X \subset \mathbb{R}^n$. Цель выбора управления $u_i \in U_i$, $i = 1, \dots, N$ заключается в осуществлении включения

$$z(p) \in X \quad (2)$$

при любой допустимой реализации помехи v_j , $j = 1, \dots, M$. На каждом шаге при построении управления используется информация о реализации помехи.

Требуется определить множество начальных положений $z(0)$, откуда возможно осуществить (2) при любых допустимых реализациях помехи.

2.2. Решение

Введем оператор T_k который каждому числу $k = 0, \dots, p$ и каждому множеству Y ставит в соответствие множество $T_k(Y)$, определяемое следующим образом. Точка $z \in T_k(Y)$ тогда и только тогда, когда для любой реализации помехи $v_j(k) \in V_j$ существует управление $u_i(k) \in U_i$ такое, что $z(k+1) \in Y$. Поэтому

$$T_k(Y) = \left(Y + \sum_{i=1}^N a_i(k)U_i \right) \overset{*}{-} \sum_{j=1}^M b_j(k)V_j. \quad (3)$$

Здесь с помощью $A \overset{*}{-} B$ обозначена геометрическая разность [5] множеств A и B из пространства \mathbb{R}^n .

Для дальнейших рассуждений потребуется следующее свойство [8]: пусть множество X является выпуклым, а числа $\sigma_i \geq 0$, $\epsilon_i > 0$, $i = 1, \dots, N$, $\delta_j \geq 0$, $\gamma_j > 0$, $j = 1, \dots, M$ удовлетворяют неравенству

$$\max_{1 \leq i \leq N} \frac{\sigma_i}{\epsilon_i} \leq \min_{1 \leq j \leq M} \frac{\delta_j}{\gamma_j}.$$

Тогда

$$\left(\left(X + \sum_{i=1}^N \epsilon_i U_i \right) \overset{*}{-} \sum_{j=1}^M \gamma_j V_j + \sum_{i=1}^N \sigma_i U_i \right) \overset{*}{-} \sum_{j=1}^M \delta_j V_j =$$

$$= \left(X + \sum_{i=1}^N (\epsilon_i + \sigma_i) U_i \right) - \sum_{j=1}^M (\gamma_j + \delta_j) V_j. \quad (4)$$

В работе [9] доказана следующая лемма.

Лемма 1. Для любого выпуклого множества K и любых чисел $\alpha_j \geq \beta_j \geq l_j \geq 0, \alpha_j \geq \beta_j + f_j, f_j \geq 0, j = 1, \dots, M$ выполнено включение

$$\begin{aligned} & \left(K + \sum_{j=1}^M (\beta_j - l_j) V_j \right) - \sum_{j=1}^M f_j V_j \supset \\ & \supset K - \sum_{j=1}^M (\alpha_j - \beta_j) V_j + \sum_{j=1}^M (\alpha_j - l_j - f_j) V_j. \end{aligned}$$

Обозначим

$$W(k) = \left(X + \sum_{i=1}^N \sum_{s=k}^p a_i(s) U_i \right) - \sum_{j=1}^M y_j(k) V_j + \sum_{j=1}^M (y_j(k) - \sum_{s=k}^p b_j(s)) V_j. \quad (5)$$

Теорема 1. Пусть $y_j(k), k = 0, \dots, p, j = 1, \dots, M$ в формуле (5) удовлетворяет условиям

$$y_j(p) = 0 \text{ и}$$

$$y_j(k+1) = y_j(k) - \max \left\{ b_j(k); y_j(k+1) \cdot \max_{1 \leq i \leq N} \left(\frac{a_i(k)}{\sum_{s=k+1}^p a_i(s)} \right) \right\}. \quad (6)$$

Тогда для множеств (5) выполнено

$$W(p) = X; W(k) \subset T_k(W(k+1)). \quad (7)$$

Доказательство. Равенство $W(p) = X$ следует из (5) и условия $y_j(p) = 0$.

Далее из (6) следует, что

$$y_j(k+1) - y_j(k) \leq - \frac{y_j(k+1) \cdot a_i(k)}{\sum_{s=k+1}^p a_i(s)},$$

$$i = 1, \dots, N, j = 1, \dots, M.$$

Поэтому

$$(y_j(k) - y_j(k+1)) \sum_{s=k+1}^p a_i(s) \geq y_j(k+1) \cdot a_i(k), \quad i = 1, \dots, N, j = 1, \dots, M.$$

Следовательно, выполняются неравенства

$$\min_{1 \leq j \leq M} \frac{y_j(k) - y_j(k+1)}{y_j(k+1)} \geq \max_{1 \leq i \leq N} \frac{a_i(k)}{\sum_{s=k+1}^p a_i(s)}. \quad (8)$$

Из (6) получим, что

$$y_j(k) - y_j(k+1) \geq b_j(k), \quad j = 1, \dots, M. \quad (9)$$

Обозначим

$$\alpha_j = y_j(k), \quad \beta_j = y_j(k+1),$$

$$l_j = \sum_{s=k}^p b_j(s), \quad f_j = b_j(k), \quad j = 1, \dots, M.$$

Тогда из неравенств (9) следует, что эти числа удовлетворяют неравенствам, содержащимся в формулировке леммы 1.

Рассмотрим множество $T_k(W(k+1))$ и покажем, что выполнено $W(k) \subset T_k(W(k+1))$. Из (3) и (5) следует, что

$$\begin{aligned} T_k(W(k+1)) &= \left(W(k+1) + \sum_{i=1}^N a_i(k)U_i \right)^* - \sum_{j=1}^M b_j(k)V_j = \\ &= \left[\left(X + \sum_{i=1}^N \sum_{s=k+1}^p a_i(s)U_i \right)^* - \sum_{j=1}^M y_j(k+1)V_j + \sum_{i=1}^N a_i(k)U_i + \right. \\ &\quad \left. + \sum_{j=1}^M (y_j(k+1) - \sum_{s=k+1}^p b_j(s))V_j \right]^* - \sum_{j=1}^M b_j(k)V_j. \end{aligned}$$

Отсюда, используя лемму 1, получим

$$\begin{aligned} T_k(W(k+1)) &\supset \\ &\left[\left(X + \sum_{i=1}^N \sum_{s=k+1}^p a_i(s)U_i \right)^* - \sum_{j=1}^M y_j(k+1)V_j + \sum_{i=1}^N a_i(k)U_i \right]^* - \\ &- \sum_{j=1}^M (y_j(k) - y_j(k+1))V_j + \sum_{j=1}^M (y_j(k) - \sum_{s=k+1}^p b_j(s) - b_j(k))V_j \supset \\ &\supset \left[\left(X + \sum_{i=1}^N \sum_{s=k+1}^p a_i(s)U_i \right)^* - \sum_{j=1}^M y_j(k+1)V_j + \sum_{i=1}^N a_i(k)U_i \right]^* - \\ &- \sum_{j=1}^M (y_j(k) - y_j(k+1))V_j + \sum_{j=1}^M (y_j(k) - \sum_{s=k}^p b_j(s))V_j. \end{aligned}$$

Применим к множеству, стоящему в правой части последнего включения, равенство (4), используя при этом неравенство (8). Тогда

$$\begin{aligned} T_k(W(k+1)) &\supset \left[\left(X + \sum_{i=1}^N \left(\sum_{s=k+1}^p a_i(s) + a_i(k) \right) U_i \right)^* - \right. \\ &- \sum_{j=1}^M (y_j(k+1) + y_j(k) - y_j(k+1))V_j + \sum_{j=1}^M (y_j(k) - \sum_{s=k}^p b_j(s))V_j = \\ &= \left[\left(X + \sum_{i=1}^N \sum_{s=k}^p a_i(s)U_i \right)^* - \sum_{j=1}^M y_j(k)V_j + \sum_{j=1}^M (y_j(k) - \sum_{s=k}^p b_j(s))V_j. \right. \end{aligned}$$

С учетом обозначений (5) получим требуемое включение $W(k) \subset T_k(W(k+1))$. Таким образом, условия (7) выполнены. Теорема доказана. \square

Построенное семейство множеств $W(k)$, $k = 0, \dots, p$ позволяет записать условия на множество начальных положений, при которых гарантируется выполнение включения (2) в момент окончания процесса управления (1).

2.3. Пример

Рассмотрим задачу управления запасами [7]

$$x(k+1) = x(k) + a(k) \cdot u - \sum_{j=1}^M b_j(k)v_j^*, \quad k = 0, 1, \dots, p. \quad (10)$$

Предполагаем, что $a(k) \geq b_j(k)$, $j = 1, \dots, M$.

Пусть $x(k) = (x^1(k); x^2(k))$, где $x^1(k)$, $x^2(k)$ — количество товаров первого и второго вида, имеющихся в наличии на складе в k -ый период времени, $x^1(k) \geq 0$, $x^2(k) \geq 0$;

$u(k) = (u^1(k); u^2(k))$, где $u^1(k)$, $u^2(k)$ — доля товара первого и второго вида, произведенного за период k , $u^1(k) \in [0; 1]$, $u^2(k) \in [0; 1]$;

$a(k)$ — максимально возможный выпуск продукции каждого вида в k -ый период времени (определяется особенностями технологического процесса);

$a(k)u^1(k)$, $a(k)u^2(k)$ — количество товара первого и второго вида, произведенного за период k .

Величина $\sum_{j=1}^M b_j(k)v_j^*$ определяется спросом на продукцию в k -й период. Информация о спросе на товар может поступать из разных источников. Предприятие получает эту информацию и на ее основании строит управление, то есть принимает решение об объеме производимой продукции в данный период времени.

Пусть к концу p -го периода требуется, чтобы выполнялось условие

$$c_1x^1(p) + c_2x^2(p) \leq \epsilon,$$

$c_1, c_2 > 0$, $x^1(p) \geq 0$, $x^2(p) \geq 0$. Это условие можно рассматривать как ограничение на ёмкость склада.

Введем управления $u_1 = (-u^1; 0)$, $u_2 = (0; -u^2)$, и $v_j = -v_j^*$. Тогда процесс управления (10) можно записать в виде (1).

Обозначим множества

$$U_1 = \{u \in \mathbb{R}^2 : u = (-u^1; 0)\},$$

$$U_2 = \{u \in \mathbb{R}^2 : u = (0; -u^2)\},$$

$$X = \{x \in \mathbb{R}^2 : c_1x^1 + c_2x^2 \leq \epsilon, x^1 \geq 0, x^2 \geq 0\}.$$

Цель процесса управления запишется в виде

$$x(p) \in X.$$

Из неравенства $a(k) \geq b_j(k)$, $j = 1, \dots, M$ следует, что

$$y_j(k) = \sum_{s=k}^p a(s), \quad k = 0, \dots, p-1, \quad y_j(p) = 0, \quad \forall j = 1, \dots, M$$

удовлетворяет условиям (6).

Тогда, подставляя $y_j(k)$ в (5), получим

$$W(k) = \left(X + \sum_{s=k}^p a(s)(U_1 + U_2) \right) - \sum_{j=1}^M \sum_{s=k}^p a(s)V_j + \sum_{j=1}^M \sum_{s=k}^p (a(s) - b_j(s)) V_j.$$

Таким образом, чтобы к концу заданного периода времени ограничения на ёмкость склада были выполнены, в начальный момент времени запас товаров на складе должен удовлетворять условию $x(0) = (x^1(0), x^2(0)) \in W(0)$.

3. Дискретная задача управления с помехой с однотипными вектограммами и выпуклой целью

3.1. Постановка задачи

Пусть задан управляемый процесс

$$z(k+1) = z(k) - a(k)u + b(k)v, \quad (11)$$

где $z \in \mathbb{R}^n$, $a(k) \geq 0$, $b(k) \geq 0$, $k = 0, \dots, p$.

Здесь $u \in X$ — значение управления, $v \in X$ — значение помехи, $X \subset \mathbb{R}^n$ является выпуклым и замкнутым. Считаем, что $p \geq 1$ — длительность процесса управления задана.

Приведем правила, описывающие управление дискретной системой. В начальный момент времени, исходя из начального положения $z(0)$, происходит реализация помехи $v(0)$ из множества X . Зная начальное положение и реализацию помехи, выбирается управление $u(0)$ из множества X . Для начального условия $z(0)$ и для выбранной допустимой пары u и v реализуется $z(1)$ по правилу (11). В следующий момент $t = 1$, исходя из значения $z(1)$, происходит реализация $v(1)$ из множества X . Зная $z(1)$ и $v(1)$, строится управление $u(1)$ из множества X и т.д.

Задано выпуклое замкнутое множество $Z \subset \mathbb{R}^n$.

Цель выбора управления $u \in X$ заключается в осуществлении включения

$$z(p) \in Z \quad (12)$$

при любой допустимой реализации помехи.

Требуется определить множество начальных положений $z(0)$, откуда возможно осуществить (12) при любых допустимых реализациях помехи.

3.2. Решение

Введем оператор T_k который каждому числу $k = 0, \dots, p$ и каждому множеству $Y \in \mathbb{R}^n$ ставит в соответствие множество $T_k(Y)$, определяемое следующим образом. Точка $z \in T_k(Y)$ тогда и только тогда, когда для любой реализации помехи $v \in X$ существует управление $u \in X$ такое, что $z(k+1) = z(k) - a(k)u + b(k)v \in Y$.

Отсюда получим, что

$$T_k(Y) = (Y + a(k)X) - b(k)X. \quad (13)$$

Обозначим

$$\Omega_m = T_m (T_{m+1} (\dots T_{p-1}(Z) \dots)), \quad m = 0, \dots, p-1. \quad (14)$$

Используя введенные обозначения получим, что множество начальных положений, откуда возможно осуществить включение (12), запишется в виде Ω_0 .

Известно [6], что для любого выпуклого замкнутого множества $M \in \mathbb{R}^n$ и для любого выпуклого компакта $X \in \mathbb{R}^n$ выполнены следующие свойства

$$(M + \epsilon X) \overset{*}{-} \epsilon X = M; \quad (15)$$

$$\left(M \overset{*}{-} \epsilon X \right) \overset{*}{-} \delta X = M \overset{*}{-} (\epsilon + \delta) X. \quad (16)$$

Здесь ϵ и δ — произвольные неотрицательные числа.

В работе [10] доказана следующая лемма.

Лемма 2. Пусть Q — выпуклое замкнутое множество в \mathbb{R}^n , $a \geq 0, b \geq 0$ — заданные числа. Тогда

$$(Q + aX) \overset{*}{-} bX = Q \overset{*}{-} b_* X + a_* X,$$

где $a_* = \max \{0; a - b\}$, $b_* = \max \{0; b - a\}$.

Теорема 2. Для любого числа $0 \leq m \leq p-1$ выполнено

$$\Omega_m = Z \overset{*}{-} \beta(m) X + \alpha(m) X,$$

где

$$\alpha(m) = \alpha(p-1) + \sum_{i=m}^{p-2} (a(i) - b(i)) \geq 0, \quad m = 0, \dots, p-2,$$

$$\alpha(p-1) = \max \{0; a(p-1) - b(p-1)\},$$

$$\beta(m) = \beta(p-1) \geq 0, \quad m = 0, \dots, p-2,$$

$$\beta(p-1) = \max \{0; b(p-1) - a(p-1)\}.$$

Доказательство. Согласно (13) и (14) множество $\Omega_{p-1} = T_{p-1}(Z) = (Z + \alpha(p-1)X) \overset{*}{-} \beta(p-1)X$.

Обозначим

$$\alpha(p-1) = \max \{0; a(p-1) - b(p-1)\} \geq 0,$$

$$\beta(p-1) = \max \{0; b(p-1) - a(p-1)\} \geq 0.$$

Покажем, что $T_{p-1} = Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X$.

Пусть $a(p-1) \geq b(p-1)$. В этом случае $\alpha(p-1) = a(p-1) - b(p-1) \geq 0, \beta(p-1) = 0$. Тогда, применяя (15), получим, что

$$(Z + \alpha(p-1)X) \overset{*}{-} \beta(p-1)X =$$

$$\begin{aligned}
 &= (Z + (a(p-1) - b(p-1))X + b(p-1)X) \overset{*}{-} b(p-1)X = \\
 &= Z + (a(p-1) - b(p-1))X = Z + \alpha(p-1)X.
 \end{aligned}$$

Показали, что в рассматриваемом случае множество $T_{p-1}(Z) = Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X$.

Пусть теперь $a(p-1) < b(p-1)$. В этом случае $\alpha(p-1) = 0, \beta(p-1) = b(p-1) - a(p-1)$. Тогда, используя (16), найдем, что

$$\begin{aligned}
 &(Z + a(p-1)X) \overset{*}{-} b(p-1)X = \\
 &= \left((Z + a(p-1)X) \overset{*}{-} a(p-1)X \right) \overset{*}{-} (b(p-1) - a(p-1))X = \\
 &= Z \overset{*}{-} (b(p-1) - a(p-1))X = Z \overset{*}{-} \beta(p-1)X.
 \end{aligned}$$

И в этом случае показали, что множество $\Omega_{p-1} = T_{p-1}(Z) = Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X$. Далее, рассмотрим множество

$$\begin{aligned}
 \Omega_{p-2} &= T_{p-2}(T_{p-1}(Z)) = T_{p-2} \left(Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X \right) = \\
 &= \left(Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X + a(p-2)X \right) \overset{*}{-} b(p-2)X.
 \end{aligned}$$

Обозначим $\alpha(p-2) = \alpha(p-1) + a(p-2) - b(p-2)$.

Снова рассмотрим два случая. Пусть сначала $a(p-2) \geq b(p-2)$. В этом случае $\alpha(p-2) \geq 0$. Значит

$$\begin{aligned}
 \Omega_{p-2} &= (T_{p-1}(Z) + a(p-2)X) \overset{*}{-} b(p-2)X = \\
 &= (T_{p-1}(Z) + (a(p-2) - b(p-2))X + b(p-2)X) \overset{*}{-} b(p-2)X = \\
 &= T_{p-1}(Z) + (a(p-2) - b(p-2))X = \\
 &= Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X + (a(p-2) - b(p-2))X = \\
 &= Z \overset{*}{-} \beta(p-1)X + (\alpha(p-1) + a(p-2) - b(p-2))X = \\
 &= Z \overset{*}{-} \beta(p-1)X + \alpha(p-2)X.
 \end{aligned}$$

Пусть теперь $a(p-2) < b(p-2)$. Тогда $\alpha(p-2) - a(p-2) + b(p-2) \geq 0$. Найдем

$$\begin{aligned}
 \Omega_{p-2} &= (T_{p-1}(Z) + a(p-2)X) \overset{*}{-} b(p-2)X = \\
 &= \left((T_{p-1}(Z) + a(p-2)X) \overset{*}{-} a(p-2)X \right) \overset{*}{-} (b(p-2) - a(p-2))X =
 \end{aligned}$$

$$\begin{aligned}
 &= T_{p-1}(Z) \overset{*}{-} (b(p-2) - a(p-2))X = \\
 &= \left(Z \overset{*}{-} \beta(p-1)X + \alpha(p-1)X \right) \overset{*}{-} (b(p-2) - a(p-2))X = \\
 &= \left(Z \overset{*}{-} \beta(p-1)X + (\alpha(p-2) - a(p-2) + b(p-2))X \right) \overset{*}{-} (b(p-2) - a(p-2))X = \\
 &= \left(Z \overset{*}{-} \beta(p-1)X + \alpha(p-2)X + (b(p-2) - a(p-2))X \right) \overset{*}{-} (b(p-2) - a(p-2))X = \\
 &= Z \overset{*}{-} \beta(p-1)X + \alpha(p-2)X.
 \end{aligned}$$

Показали, что множество $\Omega_{p-2} = Z \overset{*}{-} \beta(p-2)X + \alpha(p-2)X$.

Рассуждая аналогично, получим, что

$$\Omega_m = Z \overset{*}{-} \beta(m)X + \alpha(m)X,$$

$$\alpha(m) = \alpha(p-1) + \sum_{i=m}^{p-2} (a(i) - b(i)) \geq 0, \quad m = 0, \dots, p-2,$$

$$\alpha(p-1) = \max \{0; a(p-1) - b(p-1)\},$$

$$\beta(m) = \beta(p-1) \geq 0, \quad m = 0, \dots, p-2,$$

$$\beta(p-1) = \max \{0; b(p-1) - a(p-1)\}.$$

Теорема доказана. □

Следствие 1. Пусть $a(k) = a, b(k) = b$ для всех $k = 0, 1, \dots, p$, тогда

$$\Omega_0 = Z \overset{*}{-} \beta(p-1)X + (\alpha(p-1) + (p-1)(a-b))X.$$

Если при этом выполнено условие $a > b$, то $\Omega_0 = Z + p(a-b)X$.

Построенное семейство множеств $\Omega_m, m = 0, \dots, p-1$ позволяет определить условия на множество начальных положений, при которых гарантируется выполнение включения (12) в момент окончания процесса управления (11).

3.3. Пример

Рассмотрим задачу управления запасами [7]

$$z(k+1) = z(k) + a(k)\tilde{u} - b(k)\tilde{v}, k = 0, 1, 2, 3. \quad (17)$$

Здесь $z(k) = (z^1(k); z^2(k))$, где $z^1(k), z^2(k)$ — количество товаров первого и второго вида, имеющихся в наличии на складе в k -ый период времени, $z^1(k) \geq 0, z^2(k) \geq 0$;

$a(k)$ — максимально возможный выпуск продукции каждого вида в k -ый период времени (определяется особенностями технологического процесса);

$b(k)$ — максимально возможный спрос на продукцию каждого вида в k -ый период времени.

Предполагаем, что известны значения $a(0) = 4, a(1) = 2, a(2) = 3, b(0) = 3, b(1) = 3, b(2) = 2$. Пусть $\tilde{u}(k) = (\tilde{u}^1(k); \tilde{u}^2(k))$, где $\tilde{u}^1(k), \tilde{u}^2(k)$ — доля товара первого и второго вида, произведенного за период k , $\tilde{u}^1(k) \in [0; 1], \tilde{u}^2(k) \in [0; 1]$;

$a(k)u^1(k), a(k)u^2(k)$ — количество товара первого и второго вида, произведенного за период k ;

$\tilde{v}(k) = (\tilde{v}^1(k); \tilde{v}^2(k))$, где $\tilde{v}^1(k), \tilde{v}^2(k)$ — доля товара первого и второго вида, отгруженного со склада за период k , определяется спросом на продукцию в k -й период, $\tilde{v}^1(k) \in [0; 1], \tilde{v}^2(k) \in [0; 1]$.

Информация о спросе на товар считается известной. Предприятие получает эту информацию и на ее основании строит управление, то есть принимает решение об объеме производимой продукции в текущий момент времени.

Сделаем замену переменных

$$u^1(k) = -\tilde{u}^1(k), u^2(k) = -\tilde{u}^2(k), v^1(k) = -\tilde{v}^1(k), v^2(k) = -\tilde{v}^2(k),$$

после которой задачу (17) можно записать в виде (11).

Пусть к концу периода $p = 3$ требуется, чтобы выполнялось условие

$$1 \leq z^1(3) \leq 2, 1 \leq z^2(3) \leq 3.$$

Обозначим множества

$$X = \{x \in \mathbb{R}^2 : -1 \leq x^1 \leq 0, -1 \leq x^2 \leq 0\},$$

$$Z = \{z \in \mathbb{R}^2 : 1 \leq z^1 \leq 2, 1 \leq z^2 \leq 3\}.$$

Цель процесса управления запишется в виде $z(3) \in Z$.

Согласно теореме 2.1. множество Ω_0 запишется следующим образом

$$\Omega_0 = Z -^* \beta(0)X + \alpha(0)X,$$

где

$$\alpha(0) = \alpha(2) + \sum_{i=0}^1 (a(i) - b(i)) =$$

$$= \max\{0; a(2) - b(2)\} + a(0) - b(0) + a(1) - b(1) = 1,$$

$$\beta(0) = \max \{0; b(2) - a(2)\} = 0.$$

Окончательно получаем, что

$$\Omega_0 = Z + X.$$

Таким образом, чтобы к концу заданного периода времени ограничения на объем продукции на складе были выполнены, в начальный момент времени запас товаров на складе должен удовлетворять условию $0 \leq z^1(0) \leq 2, 0 \leq z^2(0) \leq 3$.

Заключение

Рассмотрены две модели дискретных динамических управляемых систем с помехой. В одной из них представлена дискретная задача управления, в которой вектограмма управления линейно зависит от заданных множеств. Во второй предполагается, что вектограммы управления и помехи являются однотипными множествами. В каждом случае требуется, чтобы в конечный момент времени фазовая точка содержалась в заданном множестве. При построении управления используется информация о реализации помехи. Записан оператор программного поглощения, с помощью которого сформулированы условия на множество начальных положений, при которых гарантируется выполнение требуемого включения в заданный момент времени. В качестве примера приведено решение задачи управления запасами.

Полученные результаты могут быть применены для случая, когда помеха задается в виде нечеткого числа. Как известно [11], нечеткое число полностью определяется его множеством уровня α . Рассматривая задачу для каждого множества уровня и применяя подход, предложенный в данной статье, можно построить множество начальных положений, при которых гарантируется выполнение требуемого включения, в зависимости от параметра α . Затем в виде нечеткого числа можно записать множество начальных позиций, из которых возможно осуществить цель в конечный момент времени.

Работа выполнена при финансовой поддержке РФФИ и Челябинской области в рамках научного проекта № 20-41-740027.

Литература

1. Измestьев И.В. Дискретная игровая задача с терминальным множеством в форме кольца // Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки. 2020. Т. 30, № 1. С. 18–30. DOI: 10.35634/vm200102.
2. Никитина С.А., Скорынин А.С., Ухоботов В.И. Об одной задаче управления дискретной системой // Челябинский физико-математический журнал. 2018. Т. 3, № 3. С. 311–318. DOI: 10.24411/2500-0101-2018-13304.
3. Никитина С.А., Ухоботов В.И. Дискретная динамическая задача управления с терминальным множеством в форме кольца // Вестник РАЕН. 2019. Т. 19, № 2. С. 120–121.
4. Никитина С.А., Ухоботов В.И. Об одной задаче управления запасами при наличии помехи // Челябинский физико-математический журнал. 2020. Т. 5, № 3. С. 306–315. DOI: 10.47475/2500-0101-2020-15305.

5. Понтрягин Л.С. Линейные дифференциальные игры, II // Докл. АН СССР. 1967. Т. 175, № 4. С. 764–766.
6. Понтрягин, Л.С. Линейные дифференциальные игры преследования // Мат. сб. Новая серия. 1980. Т. 112, № 3. С. 307–330.
7. Пропой А.И. Элементы теории оптимальных дискретных процессов. М.: Наука, 1973. 256 с.
8. Ухоботов В.И. К вопросу об окончании игры за первый момент поглощения // Прикладная математика и механика. 1984. Т. 48, № 6. С. 892–897.
9. Ухоботов В.И. Стабильный мост в игре с вектограммами, зависящими линейно от заданных множеств // Известия высших учебных заведений. 1988. Т. 2. С. 63–65.
10. Ухоботов В.И. Однотипные дифференциальные игры с выпуклой целью // Труды ИММ УрО РАН. 2010. Т. 16, № 5. С. 196–204.
11. Ухоботов В.И. Избранные главы теории нечетких множеств: учеб. пособие. Челябинск: Изд-во Челяб. гос. ун-та, 2011. 245 с.
12. Ухоботов В.И., Никитина С.А. Управление дискретной динамической системой с помехой // Итоги науки и техники. Серия: Современная математика и ее приложения. Тематические обзоры. 2019. Т. 168. С. 105–113. DOI: 10.36535/0233-6723-2019-168-105-113.
13. Bernardo M., Montanaro U., Olm J. M., Santini S. Model reference adaptive control of discrete-time piecewise linear systems // International Journal of Robust and Nonlinear Control. 2012. Vol. 23, no. 7. P. 709–730. DOI: 10.1002/rnc.2786.
14. Li X., Liu D., Li J., Ding D. Robust Adaptive Control for Nonlinear Discrete-Time Systems by Using Multiple Models // Mathematical Problems in Engineering. 2013. Vol. 2013. P. 1–10. DOI: 10.1155/2013/679039.
15. Li Y.-M., Li Y. Fuzzy-Model-Based Adaptive Control for a Kind of Discrete-Time Systems with Time-Delay and Disturbances // International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2014. Vol. 22, no. 3. P. 453–468. DOI: 10.1142/S0218488514500226.
16. Wang Y., Xu L. Dynamics and Control on a Discrete Multi-Inventory System // Journal of Control Science and Engineering. 2019. Vol. 2019. P. 1–7. DOI: 10.1155/2019/6926342.
17. Yongchang W., Fangyu C., Hongwei W. Inventory and Production Dynamics in a Discrete-Time Vendor-Managed Inventory Supply Chain System // Discrete Dynamics in Nature and Society. 2018. Vol. 2018. P. 1–15. DOI: 10.1155/2018/6091946.

Никитина Светлана Анатольевна, к.ф.-м.н., доцент, кафедра теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

Ухоботов Виктор Иванович, д.ф.-м.н., профессор, кафедра теории управления и оптимизации, Челябинский государственный университет (Челябинск, Российская Федерация)

DISCRETE DYNAMIC SYSTEMS WITH INTERFERENCE AND THEIR APPLICATIONS TO THE SOLUTION OF THE PROBLEM OF RESERVE CONTROL

© 2021 S.A. Nikitina, V.I. Ukhobotov

Chelyabinsk State University (st. Brat'ev Kashirinyh 129, Chelyabinsk, 454001 Russia)

E-mail: nikitina@csu.ru, ukh@csu.ru

Received: 12.03.2021

Two models of discrete dynamic controlled systems with noise are considered. In one of them a discrete control problem is presented, in which the control vectogram linearly depends on the given sets. In the second problem, it is assumed that the control and noise vectograms are of the same type. In both cases, the purpose of the choice of control is that at the moment of the end of the control process the phase point is contained in the given set. When constructing the control, it is assumed that at each discrete moment of time, information about the implementation of the interference arrives. The operator of programmed absorption is written down, with the help of which conditions are formulated for the set of initial positions under which the fulfillment of the required inclusion at a given moment of time is guaranteed. In the practical part of the work, the application of the results obtained is shown by the example of solving the problem of managing the inventory of goods in the warehouse. Replenishment of goods occurs due to its production, and the amount of shipment of goods is determined by demand. It is assumed that only the set of its values is known about the amount of demand for a product. The goal of control is that at a given moment in time, the quantity of goods satisfies certain restrictions. A lot of initial stocks of goods have been obtained, for which it is possible to fulfill the set goal for any realization of demand.

Keywords: discrete system, reserve control problem, vectograms, linearly dependent on given sets, the same type of control problem.

FOR CITATION

Nikitina S.A., Ukhobotov V.I. Discrete Dynamic Systems with Interference and Their Applications to the Solution of the Problem of Reserve Control. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 5–19. (in Russian) DOI: 10.14529/cmse210201.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Izmet'sev I.V. Discrete game problem with a terminal set in the form of a ring. *Bulletin of Udmurt University. Series: Maths. Mechanics. Computer science*. 2020. Vol. 30, no. 1. P. 18–30. (in Russian) DOI: 10.35634/vm200102.
2. Nikitina S.A., Skorynin A.S., Ukhobotov V.I. On a control problem for a discrete system. *Chelyabinsk Physical and Mathematical Journal*. 2018. Vol. 3, no. 3. P. 311–318. (in Russian) DOI: 10.24411/2500-0101-2018-13304.
3. Nikitina S.A., Ukhobotov V.I. Discrete dynamic control problem with a terminal set in the form of a ring. *Bulletin of the Russian Academy of Natural Sciences*. 2019. Vol. 19, no. 2. P. 120–121. (in Russian)
4. Nikitina S.A., Ukhobotov V.I. On one problem of reserves control in the presence of

- interference. Chelyabinsk Physical and Mathematical Journal. 2020. Vol. 5, no. 3. P. 306–315. (in Russian) DOI: 10.47475/2500-0101-2020-15305.
5. Pontrjagin L.S. Linear differential games, II. Reports of the USSR Academy of Sciences. 1967. Vol. 175, no. 4. P. 764–766. (in Russian)
 6. Pontrjagin L.S. Linear differential pursuit games. Mathematical collection. New series. 1980. Vol. 112, no. 3. P. 307–330. (in Russian)
 7. Propoj A.I. Elements of the theory of optimal discrete processes. Moscow, Nauka, 1973. 256 p.
 8. Ukhobotov V.I. On the question of ending the game in the first moment of absorption. Applied Mathematics and Mechanics. 1984. Vol. 48, no. 6. P. 892–897. (in Russian)
 9. Ukhobotov V.I. Stable bridge in a game with vectorograms depending linearly on given sets. Bulletin of higher educational institutions. 1988. Vol. 2. P. 63–65. (in Russian)
 10. Ukhobotov V.I. The same type of differential games with a convex goal. Proceedings of the Institute of Mathematics and Mechanics UB RAS. 2010. Vol. 16, no. 5. P. 196–204. (in Russian)
 11. Ukhobotov V.I. Selected chapters of fuzzy set theory. Chelyabinsk, Chelyabinsk State University, 2011. 245 p.
 12. Ukhobotov V.I., Nikitina S.A. Controlling a discrete dynamic system with interference. Results of Science and Technology. Series: Contemporary mathematics and its applications. Thematic reviews. 2019. Vol. 168. P. 105–113. (in Russian) DOI: 10.36535/0233-6723-2019-168-105-113.
 13. Bernardo M., Montanaro U., Olm J. M., Santini S. Model reference adaptive control of discrete-time piecewise linear systems. International Journal of Robust and Nonlinear Control. 2012. Vol. 23, no. 7. P. 709–730. DOI: 10.1002/rnc.2786.
 14. Li X., Liu D., Li J., Ding D. Robust Adaptive Control for Nonlinear Discrete-Time Systems by Using Multiple Models. Mathematical Problems in Engineering. 2013. Vol. 2013. P. 1–10. DOI: 10.1155/2013/679039.
 15. Li Y-M., Li Y. Fuzzy-Model-Based Adaptive Control for a Kind of Discrete-Time Systems with Time-Delay and Disturbances. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 2014. Vol. 22, no. 3. P. 453–468. DOI: 10.1142/S0218488514500226.
 16. Wang Y., Xu L. Dynamics and Control on a Discrete Multi-Inventory System. Journal of Control Science and Engineering. 2019. Vol. 2019. P. 1–7. DOI: 10.1155/2019/6926342.
 17. Yongchang W., Fangyu C., Hongwei W. Inventory and Production Dynamics in a Discrete-Time Vendor-Managed Inventory Supply Chain System. Discrete Dynamics in Nature and Society. 2018. Vol. 2018. P. 1–15. DOI: 10.1155/2018/6091946.

АВТОМАТИЧЕСКИЙ ПОДБОР ПАРАМЕТРОВ МОДЕЛИ ARIMA ДЛЯ ПРОГНОЗА КОЛИЧЕСТВА СЛУЧАЕВ ЗАРАЖЕНИЯ И СМЕРТИ ОТ COVID-19

© 2021 Т.А. Макаровских, М.С.А. Аботалеб

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: Makarovskikh.T.A@susu.ru, abotalebmostafa@yandex.ru

Поступила в редакцию: 23.11.2020

В работе исследовано применение модели ARIMA прогнозирования временных рядов для анализа открытых данных о распространении коронавирусной инфекции в ряде регионов Российской Федерации. Рассмотрена возможность применения существующих методов и алгоритмов языка программирования для статистической обработки данных R, приводятся алгоритмы подбора параметров модели ARIMA. Разработан и опубликован скрипт на языке программирования R, позволяющий осуществить с помощью стандартной библиотеки `auto.arima` прогнозирование суммарных случаев заражения и летальных исходов на выбранный промежуток времени. В работе показано, что параметры модели различны для временных рядов разной длины, для различных регионов, кроме того, параметры модели меняются с течением времени. Исследован имеющийся инструментарий языка R и показано, что существуют наборы данных, для которых он не позволяет получить параметры модели, дающие наименьшую погрешность. Исследована частота переобучения модели, приведены данные об изменении параметров модели для временных рядов разной длины. Изучение случаев ошибки автоматического подбора параметров модели является темой для дальнейших исследований. Приведена содержательная интерпретация полученных данных. Проведено сравнение прогнозов, полученных в конце октября 2020 г. и актуальных данных на середину ноября 2020 г. Показано, что полученный прогноз позволил достаточно точно предсказать суммарное число заражений и летальных исходов на 7–10 дней.

Ключевые слова: ARIMA, Covid-19, прогнозирование, скрипт, подбор параметров.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Макаровских Т.А., Аботалеб М.С.А. Автоматический подбор параметров модели ARIMA для прогноза количества случаев заражения и смерти от Covid-19 // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 20–37. DOI: 10.14529/cmse210202.

Введение

В конце 2019 года было сообщено о заражениях в провинции Хубэй (Китай) коронавирусной инфекцией нового типа. За несколько месяцев этот новый вирус вызвал глобальную пандемию коронавирусной болезни (CoViD19), которая может закончиться тяжелым острым респираторным синдромом (SARS-COV-2). Несмотря на существование известных эпидемиологических моделей, ни одна из них не может с достаточной долей точности спрогнозировать ситуацию развития эпидемии Covid-19. Исследованию вируса в настоящее время посвящено достаточно много медицинских работ, некоторые из которых не совсем корректно цитируются новостными ресурсами в погоне за сенсацией. Большинство работ является препринтами, т.е. не прошло рецензирование и проверки.

В самом начале эпидемии ежедневно обновлялась информация по долгосрочным прогнозам развития эпидемии в мире в целом и в каждой стране по отдельности. Например, эксперты Сингапурского университета технологии и дизайна (SUTD) прогнозировали, что

эпидемия в России завершится к 20 июня, а в мире — к 9 декабря, потом несколько раз переносили срок и, в конце концов, аннулировали прогнозы. Тем не менее, спустя некоторое время выяснилось, что временные ряды с данными по распространению коронавирусной инфекции пока слишком коротки для составления какого-либо адекватного долгосрочного прогноза. Построение краткосрочных прогнозов высокой точности для распространения подтвержденных случаев заболевания, а также анализа количества смертей и выздоровлений является не менее актуальной задачей.

Чтобы модель адекватно отражала истинное положение распространения инфекции, нужна достаточно точная статистика и подробный анализ внешних факторов, которые не качественно, а именно количественно влияют на распространение инфекции. Например, к таким факторам можно отнести индекс самоизоляции, плотность населения, средний возраст, вероятность заражения и др. Следует отметить, что большинство параметров, которые могли бы повысить точность прогноза, не являются очевидными.

Целью данной работы является подробное исследование работы модели ARIMA для построения краткосрочных прогнозов в условиях непредсказуемости поведения исследуемого процесса. В работе рассматривается возможность автоматического подбора параметров модели ARIMA для временных рядов, соответствующих одному и тому же процессу, протекающему в различных условиях (распространение коронавирусной инфекции в различных регионах РФ). Как показывает обзор имеющихся результатов, подобных исследований и разработок для временных рядов по распространению коронавирусной инфекции в настоящий момент найти не удалось.

Статья организована следующим образом. В первом разделе статьи рассмотрен ряд результатов, полученных российскими и зарубежными аналитиками. Во втором и третьем разделах подробно рассмотрена модель прогнозирования ARIMA и существующие методы в языке программирования для статистической обработки данных R. Приводится алгоритм подбора параметров модели ARIMA и описан созданный скрипт на языке R, для которого входными данными является вектор анализируемых параметров. В четвертом разделе исследована частота переобучения модели, приведены данные об изменении параметров модели для временных рядов разной длины. В заключении сформулированы основные полученные результаты и определены направления для дальнейших исследований.

1. Обзор состояния проблемы

В настоящий момент (конец 2020 г.) опубликовано немало работ, посвященных построению прогнозов с горизонтом 3–7 дней [12, 21, 22]. Все эти работы можно подразделить на три категории относительно используемой модели: модели SIR (классические эпидемиологические модели переходов состояний) [6, 14, 20], модели ARIMA (интегрированные модели авторегрессии и скользящего среднего) [17, 18] и модель Хольта—Уинтерса (адаптивного сглаживания) [6, 17, 19]. Одной из достаточно удачных работ можно считать статью [6], опубликованную в России, в которой авторы рассматривают указанные три модели на временном ряде с марта 2020 г. (начало эпидемии в России) по 10 мая 2020 г.

Классическая модель SIR не позволяет обеспечить высокое качество получаемых прогнозов [14, 20]. В работе [14] используется вышедшее не так давно расширение для R `covid19.analytics`, позволяющее получить полную картину по распространению вируса в любой точке земного шара. Автором разработки утверждается, что он проводит исследования на общедоступных данных, взятых из двух основных источников: «Репозиторий

данных о Covid-19 Центра системных наук и инженерии (CSSE) Университета Джона Хопкинса» [10] для данных по всему миру и США, а данные по городу Торонто берет из [11]. Пакет также предоставляет базовые инструменты и функции анализа и визуализации для исследования этих и других наборов данных, структурированных аналогичным образом. Основным недостатком данной разработки на настоящий момент является использование для прогнозирования исключительно классической модели SIR, которая дает очень большую погрешность. Еще в начале пандемии появились статьи, использующие усложненные модели SIR, учитывающие как выбывание инфицированных из-за смертности (модель SIR(D) в КНР) [20], так и инкубационный период заболевания (модель SEIR) для краткосрочного прогнозирования развития эпидемии в различных странах. Предпринимались попытки разработать модель SEIR в предположении, что развитие эпидемии начинается с подгруппы населения, характеризующейся слабым иммунным ответом, и на более поздней стадии продолжается во всем населении. Хотя это предположение кажется правдоподобным, у авторов подобных работ нет прямого подтверждения того, что неоднородность населения играет значительную роль во время распространения эпидемии, а работы до сих пор опубликованы в Интернете в статусе препринтов. Такие модели не принимают во внимание участие многих различных типов клеток, внутриклеточную регуляцию, задержку по времени в распространении вируса и клональную экспансию иммунных клеток, а также некоторые другие важные аспекты.

Данные модели хорошо подходят для более долгосрочного прогнозирования (более 7 дней). К особенностям применения эпидемиологической модели из семейства SIR(D) можно отнести то, что в ней принимаются постоянными значения коэффициентов, отвечающих за вероятность инфицирования, вероятность излечения и вероятность смерти. Однако в действительности повсеместно вырабатываются и практикуются эффективные (и не очень) меры сдерживания эпидемии (карантины, ограничение перемещения людей, масочный режим и пр.), что сказывается на изменении траектории эпидемии и, как следствие, приводит к тому, что коэффициенты подобной модели становятся переменными. В статье [6] авторы переобучают коэффициенты модели по вновь поступившим данным, что оправдано для получения краткосрочных прогнозов (до 10 дней) высокой точности.

Модели, основанные на анализе временных рядов, в частности ARIMA-модели, являются сложно настраиваемыми при проведении полного анализа, однако дают практически всегда хороший результат там, где требуется качественный прогноз на среднесрочный и краткосрочный период [6]. В работах [16, 18], как и во многих других, авторы вручную подбирают параметры для имеющихся на момент публикации временных рядов и проводят расчеты в широко используемых для статистического анализа программах (R, Gretl) для отдельных государств и регионов для фиксированных дат. Особенно много подобных работ можно найти у аналитиков из Индии и Юго-Восточной Азии. В работе [6] также приводится ряд результатов ученых из Юго-Восточной Азии и Европы, полученных подобным образом. В этой работе указываются параметры модели ARIMA, полученные в начале мая для территории России и отмечается, что для разных регионов данные параметры разнятся, более того, авторы показывают, что для разных периодов развития эпидемии данные параметры различны. В работе не приводится алгоритмов, позволяющих определить параметры в автоматическом режиме. Также хотелось бы отметить работу [13], в которой предлагается рассмотреть отношения между государствами, находящимися в одной географической зоне, для прогнозирования распространения вируса. Страны в одном географическом

ческом регионе имеют переменные с похожими значениями (поддающиеся количественной и не поддающейся количественной оценке), которые влияют на распространение вируса. Такой подход в дальнейшем может быть использован при прогнозировании распространения вируса в РФ, поскольку (как показано в разделе 4) в разных регионах имеет место различный характер распространения эпидемии. Постоянное «дообучение» модели ARIMA на разных этапах развития эпидемии является очевидным недостатком ее использования. Несмотря на то, что зачастую требуется менять не только оцениваемые параметры модели, но и число гиперпараметров модели, данный инструмент хорошо подходит для краткосрочного прогнозирования (сроком до 7 дней).

Достаточно популярным инструментом прогнозирования развития распространения коронавирусной инфекции являются адаптивные модели экспоненциального сглаживания. Данные модели также послужили достаточно распространенным инструментом для построения прогнозов для временных рядов, соответствующих развитию эпидемии в различных странах [8, 17, 19]. Как и в случае с моделью ARIMA, основным недостатком большинства представленных работ является отсутствие объяснения выбора соответствующей спецификации моделей, а также отсутствие «объяснения» подбора гиперпараметров моделей прогнозирования [6]. Отметим также работу [8] одного из соавторов данной статьи, в которой показано, что модель экспоненциального сглаживания для рассматриваемых временных рядов дает более точные результаты по сравнению с моделью ARIMA. Модель Хольта–Уинтерса фактически никак не объясняет сущность процесса развития эпидемии и исключительно ориентируется на сами данные. Таким образом, данная модель может отметить явление незначительной семидневной цикличности, связанной в первую очередь не с истинным развитием процесса инфицирования, а с графиком работы отдельных служб здравоохранения (лабораторий, производящих тестирование, а также административных служб) [6].

Таким образом, получить результаты высокой точности можно только используя динамическое оценивание изменяющихся коэффициентов. Поэтому решение данной задачи является актуальным. Проведенный анализ показывает, что большинство работ по прогнозированию случаев заражения новой коронавирусной инфекцией посвящено исследованию ручного подбора параметров модели ARIMA для данных по одному или нескольким фиксированным регионам; опубликовано очень мало работ (например, [9]), в которых предлагается использовать современные программные средства для прогнозирования временных рядов и в которых описывались бы библиотеки для анализа временных рядов указанного содержания; не удалось найти работ, посвященных особенностям прогнозирования и подбора параметров моделей для регионов Российской Федерации.

2. Алгоритм использования модели ARIMA для прогнозирования временных рядов

В качестве примеров будем использовать временные ряды по заболеваемости Covid-19 в Москве, Челябинской области и России [5] по данным с 12 марта по 27 октября 2020 г. В качестве входных данных предоставляется одномерная последовательность значений. Данные временного ряда представляют собой отсчеты значений наблюдаемых параметров в последовательные дискретные моменты времени.

Модель ARIMA состоит из трех компонентов:

- AR (авторегрессионный термин) — относится к прошлым значениям, используемым для прогнозирования следующего значения; определяется параметром p в модели авторегрессии

$$Y_t = \Theta_1 \cdot Y_{t-1} + \Theta_2 \cdot Y_{t-2} + \dots + \Theta_p \cdot Y_{t-p} + \varepsilon_t,$$

который определяется по PACF (partial auto-correlation function) — «частная корреляционная функция» между Y_t и Y_{t-k} при исключении влияния $Y_{t-1}, \dots, Y_{t-k+1}$.

- MA (скользящее среднее) — используется для определения количества прошлых ошибок прогноза, используемых для прогнозирования будущих значений; определяется параметром q , получаемого из ACF (auto-correlation function, автокорреляционная функция)

$$\rho_k = \frac{\text{cov}\{Y_t, Y_{t-k}\}_t}{\text{var}\{Y_t\}_t},$$

$$y_t = \varepsilon_t + \alpha \cdot \varepsilon_{t-1} + \dots + \alpha_q \cdot \varepsilon_{1-q},$$

где ε_t — белый шум, всегда являющийся стационарным процессом¹. Скользящее среднее показывает наличие колебаний в ряду. Чем выше значение скользящего среднего, тем выше вероятность колебаний.

- I (интегрирующий член) — если ряд не является стационарным, то находится его разность порядка d , которая является стационарным рядом. Для проверки стационарности ряда используются расширенный тест Дики—Фулера (ADF)² тест Квятковского—Филлипса—Шмидта—Шина (Kwiatkowski—Phillips—Schmidt—Shin, KPSS)³. Эти же тесты позволяют определить параметр d модели.

Приведем алгоритм построения модели ARIMA.

Алгоритм ARIMA-Construct

Шаг 1. Загрузить входные данные в вектор TS.

Шаг 2. Проверить, является ли временной ряд TS стационарным. Если TS является стационарным, перейти на шаг 4. Если ряд не является стационарным, то находится его разность порядка d , которая является стационарным рядом.

Шаг 3. Положить $d = 0$.

Шаг 4. Построить графики ACF и PACF для определения входных параметров модели ARIMA.

Шаг 5. С помощью графиков ACF и PACF определить значения p и q для модели ARIMA.

Шаг 6. Выполнить подгонку параметров p и q для идентификации модели ARIMA.

Шаг 7. Спрогнозировать значения на проверочном наборе TS: будущие значения.

Шаг 8. Рассчитать среднеквадратическое отклонение для сравнения прогнозов и фактических значений.

Конец алгоритма

Рассмотрим применение данного алгоритма на временном ряде, соответствующем количеству заражений Covid-19 в день по Челябинской области на конец октября 2020 года. В проведенном эксперименте параметры модели ARIMA подбирались с помощью скриптов

¹Белым шумом называется процесс, имеющий постоянное математическое ожидание, постоянную дисперсию и нулевую, для всех, кроме нулевого лага, автоковариационную функцию.

²Методика, которая используется в прикладной статистике и эконометрике для анализа временных рядов для проверки на стационарность.

³Критерий, используемый для проверки на стационарность наблюдаемого временного ряда.

на языке R. Пример подбора параметров для Челябинской области приведен в [1]. В таблицах приведена описательная статистика по зарегистрированным случаям в день (табл. 1) и летальным случаям (табл. 2). Термины, используемые в таблице, являются общеупотребительными в статистике [7].

Таблица 1

Описательная статистика по заражениям Covid-19 в некоторых регионах РФ

Регион	Среднее	Min	Max	Станд. откл.	Асимм.	Экспесс
Челябинская обл.	7720.7740	0	18 393	6 300.5310	0.0926	1.4626
г. Москва	18 8832.7000	21	405 352	111 203.5000	-0.4602	2.1716
Московская обл.	45 981.6900	4	84 247	27 874.2200	-0.5690	1.7950
г. Санкт-Петербург	23 544.6700	1	57 515	16 584.0400	-0.0078	1.8397
Свердловская обл.	14 259.9300	0	34 782	11 672.7100	0.1144	1.5033
Нижегородская обл.	17 059.3400	1	39 636	12 480.2500	-0.0348	1.6364
Респ. Крым	1 643.7910	0	8 533	2 069.1200	1.5772	4.6731

Таблица 2

Описательная статистика по летальным случаям Covid-19 в некоторых регионах РФ

Регион	Среднее	Min	Max	Станд. откл.	Асимметрия	Экспесс
Челябинская обл.	65.1609	0	159	53.3066	0.0030	1.5226
г. Москва	3 180.0090	0	6503	2 054.9040	-0.4150	1.6976
Московская обл.	752.5633	0	1549	524.2495	-0.2676	1.5088
г. Санкт-Петербург	1 359.0350	0	3627	1 224.3110	0.2248	1.5364
Свердловская обл.	229.2533	0	743	245.7027	0.6538	1.9212
Нижегородская обл.	271.2882	0	721	228.9393	0.2134	1.6182
Респ. Крым	22.6044	0	122	30.8129	1.7321	5.1280

Несмотря на то, что ARIMA является очень мощной моделью для прогнозирования временных рядов, процессы подготовки данных и настройки параметров в итоге отнимают немало времени: требуется сделать ряд стационарным, определить значения p и q с помощью графиков. Кроме того, как было отмечено выше, при анализе данных по пандемии Covid-19 целесообразно использовать динамические параметры моделей, что требует постоянного переобучения модели для каждого отдельно взятого набора данных. Данный факт является причиной появления большого числа исследований, посвященных подбору параметров модели для временных рядов, разнообразных по продолжительности и происхождению. Вопрос о частоте переобучения модели ARIMA для временных рядов по Covid-19, соответствующих различным регионам земного шара, является открытой задачей.

3. Использование функции `auto.arima` в языке R для автоматического подбора параметров модели

Поскольку процесс подбора параметров модели ARIMA является достаточно трудоемким и требует анализа графиков, целесообразно использовать модель AutoARIMA, которая

автоматизирует выполнение шагов 2–5 алгоритма ARIMA-Construct. Приведем алгоритм, использующий автоматический подбор параметров.

Алгоритм AutoARIMA

Шаг 1. Загрузить входные данные (временной ряд) в вектор TS.

Шаг 2. FitAutoARIMA(TS). // Подогнать модель для временного ряда TS.

Шаг 3. Предсказать значения для проверочного набора.

Шаг 4. Проверить производительность модели, используя прогнозные значения в сравнении с фактическими.

Конец алгоритма

Легко видеть, что данный алгоритм полностью обходит выбор функций p и q .

Для анализа временных рядов в R может быть полезна функция `auto.arima`, которая в качестве входных данных принимает одномерные временные ряды и использует алгоритм, который объединяет тесты для подбора лучшей модели ARIMA. Очевидно, что поскольку `auto.arima` — автоматизированная процедура, то ее использование связано с рядом предположений, проверок и противопоставлений. На практике рекомендуется проверка полученных параметров автоматического процесса на соответствие заданным критериям, при необходимости разрешается изменить их. Рассмотрим работу функции `auto.arima` для временных рядов, предложенных в предыдущем разделе. Получим следующие результаты, приведенные в [3, 4]. Таким образом, процесс прогнозирования можно представить в виде следующей структурной схемы (см. рис. 1). Здесь левая ветвь структурной схемы соответствует алгоритму ARIMA-Construct, а правая — алгоритму AutoARIMA. Отметим, что для выполнения всех шагов указанных алгоритмов в R имеются стандартные библиотеки и функции, которые требуется вызвать в необходимой последовательности.

Приведем результаты использования рассмотренного скрипта для временного ряда с данными по суммарному количеству заражений по Челябинской области. Имеем, что по данным до 27 октября включительно лучшая модель для Челябинской области — ARIMA(4,2,0). Графический анализ остатков модели приведен на рис. 2.

Напомним, что остатки в модели временных рядов — это то, что остается после подбора модели. Таким образом, остатки равны разнице между наблюдениями и соответствующими подобранными значениями. Остатки используются для проверки, адекватности реакции модели на информацию о данных. Хороший метод прогнозирования дает остатки со следующими свойствами [15]:

1. Остатки не коррелированы. Если есть корреляции между остатками, значит, в остатках остается информация, которую следует использовать при вычислении прогнозов.
2. Остатки имеют нулевое среднее. Если остатки имеют среднее значение, отличное от нуля, то прогнозы смещены.
3. Остатки имеют постоянную дисперсию.
4. Остатки имеют нормальное распределение.

Рисунок 2 состоит из трех частей, первая часть (верхний график) представляет собой остатки от прогнозирования случаев заражения Covid-19 в Челябинске с использованием наивного метода. Вторая часть (левый нижний рисунок) представляет собой ACF остатков от наивного метода, примененного к случаям инфекции Covid-19. Отсутствие корреляции говорит о том, что получаемые прогнозы имеют достаточно высокое качество. И третья часть (правый нижний график) представляет собой гистограмму остатков от наивного метода, примененного к случаям заражения Covid-19 в Челябинске. Правый хвост представ-

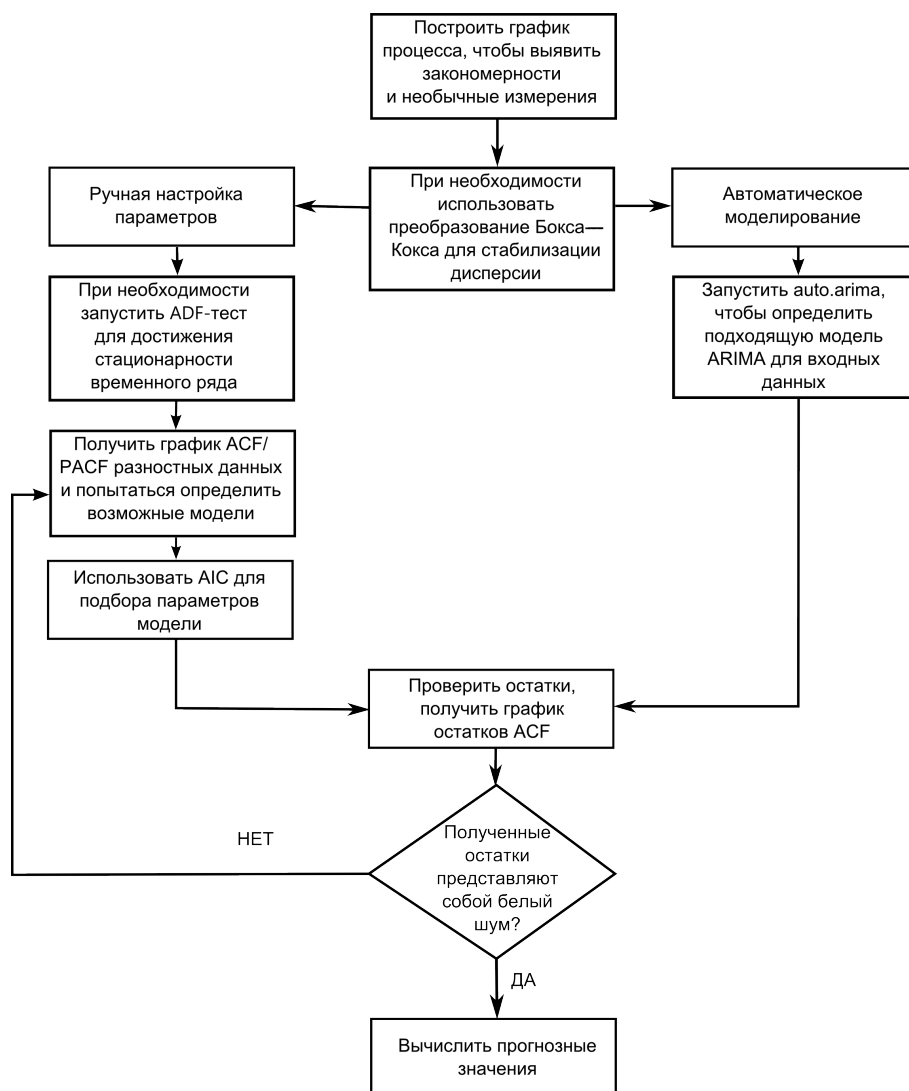


Рис. 1. Структурная схема для алгоритма выбора способа моделирования

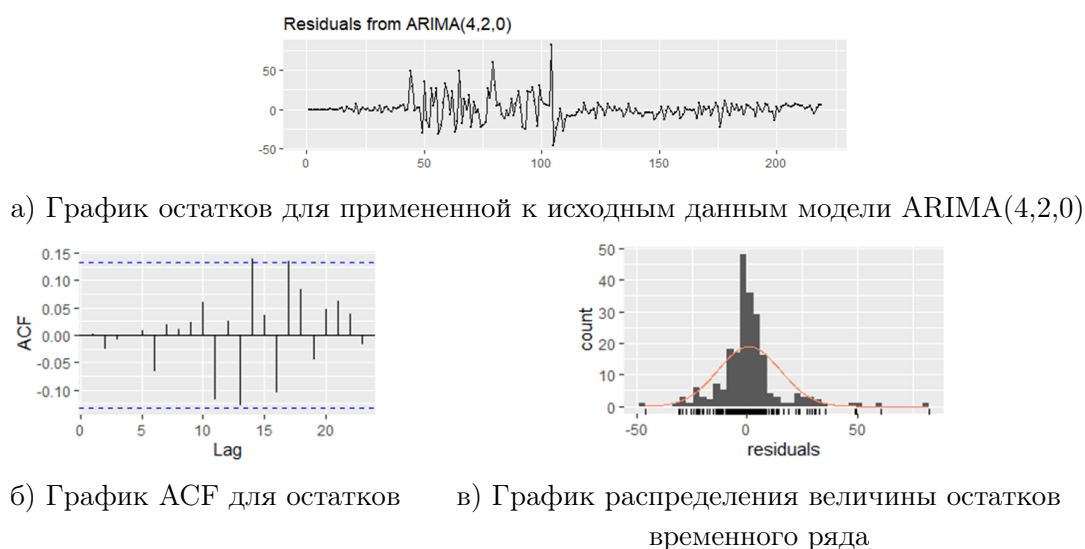


Рис. 2. Графический анализ остатков для модели ARIMA(4,2,0), использованной для прогнозирования числа случаев заражения Covid-19 по Челябинской области

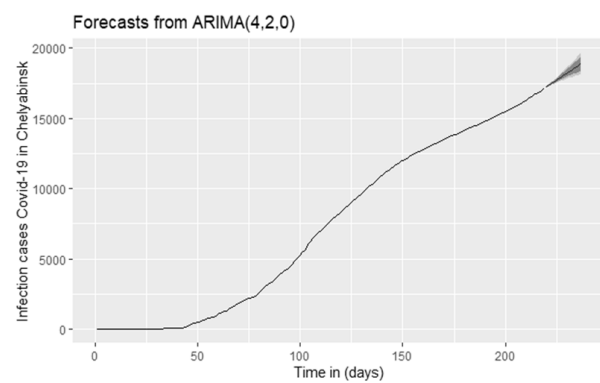


Рис. 3. Прогноз суммарного количества заражений Covid-19 по Челябинской области

ляется слишком длинным для нормального распределения в случаях заражения Covid-19. Эти графики показывают, что наивный метод дает прогнозы, которые, по-видимому, учитывают всю доступную информацию. Среднее значение остатков близко к нулю, и в рядах остатков нет значительной корреляции. Временной график остатков показывает, что вариация остатков остается практически неизменной для исторических данных, за исключением одного выброса, и поэтому остаточную дисперсию можно рассматривать как постоянную. Это также можно увидеть на гистограмме остатков. Гистограмма предполагает, что распределение остатков может не быть нормальным — правый хвост кажется слишком длинным, даже в случае игнорирования выброса. Следовательно, прогнозы этого метода, вероятно, будут довольно хорошими, но интервалы прогнозирования, которые вычисляются с учетом нормального распределения, могут быть неточными. Поэтому проводится тест Льюнга—Бокса⁴, чтобы убедиться, что нет автокорреляции из-за белого шума⁵.

Существует множество методов проверки остатков, удобно упакованных в одну функцию R `checkresiduals()`, которая создаст временной график, график ACF и гистограмму остатков (с наложенным нормальным распределением для сравнения) и проведет тест Льюнга—Бокса с правильными степенями свободы. В рассмотренном случае [3, 4] полученные с помощью программа остатки выглядят как белый шум, что позволяет использовать полученную модель для прогноза. Так, прогноз заболеваемости в Челябинской области на 10 дней вперед приведен на рис. 3. Аналогичные результаты можно получить и для других регионов Российской Федерации.

Разработанный скрипт можно использовать и для прогнозирования случаев смерти в рассмотренных регионах. Отметим, что в этом случае параметры моделей зачастую отличаются от параметров моделей, использованных для прогнозирования суммарного количества заражений (рис. 4). В табл. 3 приведем результаты анализа точности полученных прогнозов. Отметим, что при правильном подборе параметров модели удается с высокой точностью спрогнозировать как суммарное количество случаев заражения, так и суммарное число летальных случаев. Так, при прогнозировании случаев заражений и летальных исхо-

⁴Тест Льюнга—Бокса — статистический критерий, предназначенный для нахождения автокорреляции временных рядов. Вместо тестирования на случайность каждого отдельного коэффициента, он проверяет на отличие от нуля сразу несколько коэффициентов автокорреляции. Обычно тест используется при построении моделей ARIMA. При этом следует иметь в виду, что данное тестирование применяется к остаткам полученной модели, а не к исходным данным.

⁵Белым шумом называется процесс, имеющий постоянное математическое ожидание, постоянную дисперсию и нулевую, для всех, кроме нулевого лага, автоковариационную функцию.

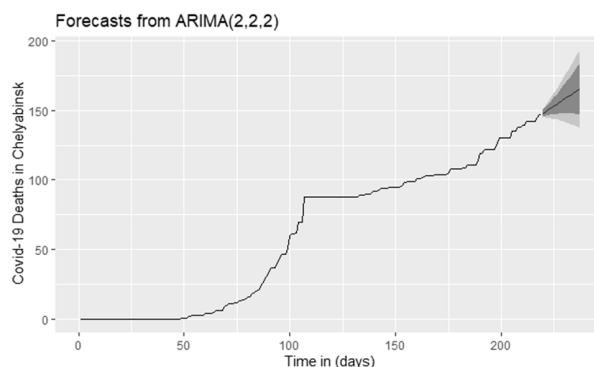


Рис. 4. Полученные лучшие параметры модели ARIMA для Челябинской обл. и результаты прогнозирования суммарного числа летальных случаев Covid-19

Таблица 3

Анализ погрешности прогнозных моделей для Челябинской области

Дата	Случаи заражения			Случаи смерти		
	Факт	Прогноз	Ошибка %	Факт	Прогноз	Ошибка %
17.10.2020	17 223	17 223.78	0.004529	147	147.9774	0.664898
18.10.2020	17 325	17 322.90	0.012121	147	149.2462	1.528027
19.10.2020	17 427	17 420.61	0.036667	152	150.1797	1.197566
20.10.2020	17 526	17 517.87	0.046388	152	151.1371	0.567697
21.10.2020	17 630	17 615.96	0.079637	155	152.1731	1.823806
22.10.2020	17 740	17 714.48	0.143856	155	153.1800	1.174194
23.10.2020	17 857	17 813.21	0.245226	155	154.1754	0.532
24.10.2020	17 986	17 911.82	0.412432	159	155.1815	2.401572
25.10.2020	18 122	18 010.17	0.617095	159	156.1875	1.768868
26.10.2020	18 255	18 108.47	0.802684	159	157.1907	1.137925
27.10.2020	18 393	18 206.77	1.012505	159	158.1948	0.506415
MAPE (%)			0.310285	MAPE (%)		1.209361

дов от Covid-19 по Челябинской области средняя абсолютная процентная ошибка (MAPE) не превышает 1.3%. Здесь и далее в статье в столбце табл. «Прогноз» встречаются вещественные значения, поскольку модель с автоматическим подбором параметров использует непрерывные данные. Округление осуществляется на окончательном этапе при оформлении полученной таблицы и представлении данных для общественности. Для анализа ошибок округлять полученные значения не целесообразно.

Сравним среднюю абсолютную процентную ошибку (MAPE) прогнозов, полученных с помощью `auto.arima`, и прогнозов по данным из работы [6]. Напомним, что модель ARIMA в указанной работе постоянно переобучалась, подбор параметров не осуществлялся автоматически. Будем рассматривать только последний набор данных из работы [6], использующий данные до 10 мая 2020 года включительно. В результате получим следующие значения погрешностей прогноза на неделю (см. табл. 4).

Из таблицы видно, что качество прогнозирования с помощью автоматического подбора параметров сравнимо с использованием моделей с ручным подбором параметров. Отме-

Таблица 4

Сравнение погрешностей прогноза для модели ARIMA с автоматическим подбором параметров и модели, описанной в работе [6] по РФ

Дата	Случаи заражения (модель auto.arima)			Случаи заражения (модель ARIMA, параметры из [6])		
	Факт	Прогноз	Ошибка (%)	Факт	Прогноз	Ошибка (%)
11.05.2020	221 344	220 613.6	0.33	221 344	219 615	0.78
12.05.2020	232 243	231 632.9	0.263	232 243	230 661	0.68
13.05.2020	242 271	242 652.2	0.157	242 271	241 780	0.20
14.05.2020	252 245	253 671.5	0.566	252 245	252 946	0.28
15.05.2020	262 843	264 690.8	0.703	262 843	264 194	0.51
16.05.2020	272 043	275 710.2	1.348	272 043	275 541	1.29
17.05.2020	281 752	286 729.5	1.767	281 752	286 958	1.85
		MAPE (%)	0.733		MAPE (%)	0.799

тим, что получить высокую точность прогнозов возможно только в случае рассмотрения каждого региона по отдельности. Подобрать наиболее подходящую модель для прогнозирования указанных прогнозов для всей страны в целом с такой же высокой точностью оказалось невозможным. Причиной этому является зависимость скорости распространения инфекции от плотности населения, индекса самоизоляции, соблюдения мер безопасности и прочих факторов, значительно различающихся для каждого региона.

Кроме того, несмотря на заметную экономию времени для подбора параметров открытым остается вопрос о частоте переобучения модели ARIMA для анализа временных рядов по Covid-19. Чтобы получить высокий уровень точности и снизить частоту ошибок, лучше всего выполнять этот анализ еженедельно, потому что некоторые изменения, которые менее заметны в случаях заражения и смерти, могут быть реакцией на предпринятые государством либо регионом меры по сдерживанию распространения инфекции. Прочие факторы также могут привести к снижению уровня точности и увеличению ошибки, поэтому целесообразно осуществлять прогнозирование не чаще раза в неделю.

4. Частота переобучения модели ARIMA при анализе данных по Covid-19

В данной работе использованы наборы данных [10, 11], а также официальные данные по регионам РФ, предоставляемые на портале [5]. Одним из результатов работы [6] является настройка моделей, позволяющая (на май 2020 года) получать краткосрочные прогнозы достаточно высокой точности (ошибка порядка 0.3 %). Так, в период 11–17 мая 2020 года средняя абсолютная процентная ошибка по модели Хольта–Уинтерса 0.84 %, по модели SIR(D), дающей самый пессимистичный прогноз, — 1.1 %, по модели ARIMA — 0.8 %. Будем в дальнейшем использовать данные значения как ориентир.

Рассмотрим подбор параметров модели ARIMA при загрузке в разработанный скрипт входного вектора с данными на последний день каждого месяца (с апреля по октябрь) для рассматриваемых регионов (табл. 5).

Отметим, что используется стандартное кодирование ARIMA (p, d, q), где параметры заменяются целочисленными значениями. Каждый из этих компонентов явно указан в мо-

Таблица 5

Параметры моделей ARIMA при загрузке временного ряда на конец каждого месяца
(с апреля по октябрь 2020 г.)

Регион		30.04	31.05	30.06	31.07	31.08	30.09	31.10
Челябинская обл.	Заражения	(0,2,0)	(2,2,3)	(4,2,0)	(4,2,0)	(4,2,0)	(4,2,0)	(4,2,0)
	Смерти	(0,0,0)	(0,0,1)	(4,2,0)	(1,2,3)	(1,2,3)	(1,2,3)	(1,2,3)*
г. Москва	Заражения	(0,2,0)	(1,2,0)	(1,2,0)	(1,2,0)	(1,2,0)	(2,2,2)	(3,2,2)*
	Смерти	(2,2,0)	(0,2,2)	(0,2,2)	(1,2,1)	(1,2,1)	(1,2,1)	(2,2,2)*
Респ. Крым	Заражения	(0,2,1)	(0,2,1)	(0,2,5)	(2,2,3)	(0,2,5)	(0,2,5)	(5,2,0)*
	Смерти	(0,0,0)	(0,1,2)	(0,2,1)	(0,2,1)	(0,2,1)	(2,2,2)	(3,2,1)*
Московская обл.	Заражения	(0,2,1)	(1,2,0)	(1,2,0)	(1,2,2)	(1,2,2)	(1,2,2)	(1,2,2)
	Смерти	(0,2,5)	(2,2,3)	(0,2,5)	(0,2,5)	(0,2,5)	(0,2,5)	(0,2,5)
г. С.Петербург	Заражения	(0,2,1)	(4,2,0)	(4,2,0)	(4,2,0)	(0,2,5)	(0,2,5)	(0,2,5)
	Смерти	(0,2,4)	(0,2,1)	(0,2,3)	(4,2,0)	(1,2,3)	(1,2,3)	(1,2,3)*
Свердловская обл.	Заражения	(1,2,0)	(0,2,2)	(0,2,1)	(0,2,1)	(0,2,1)	(2,2,3)	(2,2,3)
	Смерти	(0,1,0)	(0,2,1)	(5,2,0)	(2,2,3)	(2,2,3)	(2,2,3)	(1,2,2)
Нижегородская обл.	Заражения	(0,2,1)	(1,2,3)	(5,2,0)	(0,2,5)	(0,2,5)	(0,2,5)	(0,2,5)
	Смерти	(0,2,1)	(1,2,1)	(5,2,0)	(1,2,2)	(2,2,3)	(0,2,1)	(0,2,1)

дели в виде коэффициента. При изучении временных рядов по коронавирусной инфекции в регионах было определено, что для них имеет место стационарность второго порядка (второй параметр для всех полученных моделей равен 2). Принципиальное различие между моделями AR и MA основано на корреляции между объектами временных рядов в разные моменты времени. Ковариация между Y_t и Y_{t-n} равна нулю для моделей MA. Однако корреляция между Y_t и Y_{t-n} постепенно уменьшается по мере увеличения n в модели AR. Это означает, что модель скользящего среднего (MA) не использует предыдущие прогнозы для прогнозирования будущих значений при использовании ошибок из предыдущих прогнозов. В то время как модель авторегрессии (AR) использует предыдущие прогнозы для прогнозирования будущих значений. Данный метод позволяет узнать, информацию по скольким дням из прошлого необходимо использовать для получения прогноза будущих значений. Параметр определяется путем вычисления автокорреляции между количеством случаев в настоящий момент и по прошедшим дням, пока не будет достигнуто максимальное количество дней в прошлом, которые модель ARIMA может использовать с наибольшим весом. Например, AR (3) означает, что предсказанный новый день зависит от трех предшествующих ему дней. Так, модель ARIMA Санкт-Петербурга для случаев (0,2,5) 27 октября 2020 г. и степень MA равна 5, что означает, что в октябре есть колебания в количестве выявленных случаев.

Заметим, что в последнем столбце табл. 5 ряд значений отмечено звездочкой, это данные, для которых с помощью модели `auto.arima` не удалось получить наименьшую погрешность прогнозирования. Из табл. 5 видно, что имеет место изменение параметров модели с течением времени и накоплением дополнительной информации. Вопрос о скорости изменения параметров и «переломных» моментах, в которые имеет место смена параметров прогнозной модели остается открытым и является темой дальнейших исследований.

Рассмотрим интерпретацию полученных результатов для Челябинской области. Аналогичные рассуждения можно провести и для других регионов. Ожидается, что в ближайший период в Челябинске продолжится рост случаев заражения, количество случаев заражения через неделю превысит 20 тысяч, поэтому необходимо принять дополнительные меры для

предотвратить такой рост случаев инфицирования. В ближайшее время ожидается небольшое увеличение ежедневной смертности в Челябинске, но в ноябре 2020 года оно в сумме не превысит 200 случаев. Как следует из имеющейся статистики [5] на 19 ноября 2020 года, количество заражений равно 22627, что соответствует полученному прогнозу. Количество летальных исходов — 193. Таким образом, спрогнозированный предел будет достигнут уже 20 ноября. На дату подачи статьи разработанная модель позволяет сделать следующий прогноз: количество случаев заражения составит более 220 случаев в сутки, а смертность достигнет 15 случаев в сутки. Отметим, что полученные результаты позволяют судить о достаточно хорошем качестве краткосрочных прогнозов с использованием модели ARIMA. Таких прогнозов зачастую может быть достаточно для принятия управленческих решений.

Заключение

Пандемия коронавируса стала глобальным вызовом для науки: не только для медицины, но и для математики и статистики. Требуется разработка новых подходов к моделированию и прогнозированию временных рядов в условиях непредсказуемости поведения исследуемого процесса. Говорить о построении модели, которая адекватно описывала бы процесс распространения коронавирусной инфекции, в настоящий момент преждевременно, поскольку не прошло еще года с начала пандемии, нет возможности оценить возможный сезонный характер данных.

К основным полученным результатам можно отнести следующие.

1. Показано, что параметры модели различны для различных регионов, кроме того, параметры модели меняются с течением времени. Исследование закономерностей изменения параметров модели остается открытой задачей. В работе рассматривается возможность автоматического подбора параметров модели ARIMA для временных рядов, соответствующих одному и тому же процессу, протекающему в различных условиях (распространение коронавирусной инфекции в различных регионах РФ).

2. Разработан скрипт на языке программирования R, позволяющий осуществить с помощью стандартной библиотеки `auto.arima` прогнозирование суммарных случаев заражения и летальных исходов на выбранный промежуток времени в указанном пользователем регионе или государстве. Показано, что автоматический подбор параметров в ряде случаев не позволяет выбрать, действительно, лучшую модель. Исследование случаев ошибки автоматического подбора параметров модели является темой для дальнейших исследований. Разработанный скрипт включен в общую библиотеку `epidemic.ta` скриптов для прогнозирования временных рядов по Covid-19.

В качестве направлений дальнейших исследований можно выделить изучение и выявление особенностей использования прочих классических моделей прогнозирования временных рядов, пополнение библиотеки `epidemic.ta` [2], позволяющей получать и исследовать прогнозы с помощью различных методов, а также рассмотрение применимости неклассических (квазилинейных) моделей для построения прогнозов развития коронавирусной инфекции и применимости полученных моделей для прогнозирования временных рядов схожей структуры для прочих прикладных задач, например, для задач экономики, логистики, распространения информации в социальных сетях и пр.

Статья выполнена при поддержке Правительства РФ (Постановление № 211 от 16.03.2013 г.), Соглашение № 02.A03.21.0011 и Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

Литература

1. Аботалев М.С.А., Макаровских Т.А. Ручной подбор параметров модели ARIMA для прогнозирования летальных случаев Covid-19 в Челябинский области. URL: <https://rpubs.com/abotalebmostafa/710535> (дата обращения: 09.01.2021).
2. Аботалев М.С.А., Макаровских Т.А. New package «Epidemic.ta» for forecasting Covid-19 infection cases apply Example For forecasting infection cases in the USA. URL: <https://rpubs.com/abotalebmostafa/711262> (дата обращения: 09.01.2021).
3. Аботалев М.С.А., Макаровских Т.А. Research opportunities automatic parameter selection arima models for short term forecast cases Covid-19. URL: <https://rpubs.com/abotalebmostafa/710470> (дата обращения: 09.01.2021).
4. Аботалев М.С.А., Макаровских Т.А. Using auto arima for forecasting Deaths of COVID-19 in Chelyabinsk. URL: <https://rpubs.com/abotalebmostafa/710528> (дата обращения: 09.01.2021).
5. Коронавирус: статистика. URL: <https://yandex.ru/covid19/stat> (дата обращения: 09.01.2021).
6. Лакман И.А., Агапитов А.А., Садикова Л.Ф. и др. Возможности математического прогнозирования коронавирусной инфекции в Российской Федерации // Артериальная гипертензия. 2020. Т. 26, № 3. С. 288–294. DOI: 10.18705/1607-419X-2020-26-3-288-294.
7. Эверитт Б. Большой словарь по статистике. СПб.: Проспект, 2012. 736 с.
8. Abotaleb M.S.A. Predicting COVID-19 Cases using Some Statistical Models: An Application to the Cases Reported in China Italy and USA // Academic Journal of Applied Mathematical Sciences. 2020. Vol. 6, no. 4. P. 32–40. DOI: 10.32861/ajams.64.32.40.
9. Chaurasia V., Pal S. COVID-19 Pandemic: ARIMA and Regression Model-Based Worldwide Death Cases Predictions // SN Computer Science. 2020. Art. 1:288. DOI: 10.1007/s42979-020-00298-6.
10. COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. URL: <https://github.com/CSSEGISandData/COVID-19> (дата обращения: 23.11.2020).
11. COVID-19: Status of Cases in Toronto. URL: <https://www.toronto.ca/home/covid-19/covid-19-latest-city-of-toronto-news/covid-19-status-of-cases-in-toronto/> (дата обращения: 23.11.2020).
12. Elie R., Hubert E., Turinici G. Contact rate epidemic control of COVID-19: an equilibrium view // Mathematical Modelling of Natural Phenomena. 2020. Vol. 15, art. 35. DOI: 10.1051/mmnp/2020022.
13. Hernandez-Matamorosb A., Fujitaa H., Hayashib T., Perez-Meana H. Forecasting of COVID-19 per regions using ARIMA models and polynomial functions // Applied Soft Computing Journal. 2020. Vol. 96, art. 106610. DOI: 10.1016/j.asoc.2020.106610.
14. Hussain N., Li B. Using R-studio to examine the COVID-19 Patients in Pakistan Implementation of SIR Model on Cases // International Journal of Scientific Research in Multidisciplinary Studies. 2020. Vol. 6, no. 8. P. 54–59. DOI: 10.13140/RG.2.2.32580.04482.
15. Hyndman R.J., Athanasopoulos G. Forecasting: principles and practice. OTexts, 2018. 384 p.

16. Kumar M., Gupta S., Kumar K., Sachdeva M. Spreading of Covid-19 in India, Italy, Japan, Spain, UK, US: A Prediction Using ARIMA and LSTM Model // Digital Government: Research and Practice. 2020. Vol. 1, no. 4, art. 24. DOI: 10.1145/3411760.
17. Panda M. Application of ARIMA and Holt–Winters forecasting model to predict the spreading of COVID-19 for India and its states. DOI: 10.1101/2020.07.14.20153908 (дата обращения: 10.02.2021).
18. Perone G. ARIMA forecasting of COVID-19 incidence in Italy, Russia, and the USA. DOI: 10.2139/ssrn.3612402 (дата обращения: 10.02.2021).
19. Shokeralla A.A.A., Sameeh F.R.I., Musa A.G.M., Zahrani S. Prediction the daily number of confirmed cases of covid-19 in Sudan with arima and holt winter exponential smoothing // International Journal of Development Research. 2020. Vol. 10, no. 8. P. 39408–39413 DOI: 10.37118/ijdr.19811.08.2020.
20. Sun D., Duan L., Xiong J., Wang D. Modelling and forecasting the spread tendency of the COVID-19 in China // BMC Infectious Diseases. DOI: 10.21203/rs.3.rs-26772/v1 (дата обращения: 10.02.2021).
21. Volpert V., Banerjee M., d’Onofrio A., Lipniacki T., Petrovskii S., Tran V.Ch. Coronavirus: scientific insights and societal aspects // Mathematical Modelling of Natural Phenomena. 2020. Vol. 15, art. E2. DOI: 10.1051/mmnp/2020010.
22. Volpert V., Banerjee M., Petrovskii S., Volpert V. On a quarantine model of coronavirus infection and data analysis // Mathematical Modelling of Natural Phenomena. 2020. Vol. 15, art. 24. DOI: 10.1051/mmnp/2020006.

Макаровских Татьяна Анатольевна, д.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Аботалёв Мостафа Салахелдин Абделсалам, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

AUTOMATIC SELECTION OF ARIMA MODEL PARAMETERS TO FORECAST COVID-19 INFECTION AND DEATH CASES

© 2021 Т.А. Makarovskikh, М.С.А. Abotaleb

South Ural University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: Makarovskikh.T.A@susu.ru, abotalebmostafa@yandex.ru

Received: 23.11.2020

In our paper we explore the use of the ARIMA model for forecasting time series for the analysis of open data on the spread of the coronavirus infection in a number of the Russian Federation regions. The possibility of using the existing methods and algorithms of R programming language is considered, algorithms for selecting the parameters of the ARIMA model are presented. We have developed and uploaded the script in R programming language, which allows using the standard library `auto.arima` to predict the total cases of infection and deaths for a selected period. The paper shows that the parameters of the model are different for time series of different lengths, for different regions; in addition, the parameters of the model change over time. The available toolkit of the R language is investigated and it is shown that there are data sets for which it does not allow obtaining the parameters of the model that gives the smallest error. The frequency of model retraining is investigated, data on changes in the model parameters for time series of different lengths are presented. Investigation of cases of errors in automatic selection of model parameters is a topic for further research. We have presented a meaningful interpretation of the data obtained. A comparison of the forecasts obtained at the end of October, 2020 and actual data for the middle of November, 2020 is carried out. We have shown that the obtained forecast made it possible to accurately predict the total number of infections and deaths for 7–10 days for any further period.

Keywords: ARIMA, Covid-19, forecasting, script, parameters selection.

FOR CITATION

Makarovskikh T.A., Abotaleb M.S.A. Automatic Selection of ARIMA Model Parameters to Forecast COVID-19 Infection and Death Cases. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 20–37. (in Russian) DOI: 10.14529/cmse210202.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Abotaleb M.S.A., Makarovskikh T.A. Manual selection of ARIMA model parameters for predicting death cases of Covid-19 in the Chelyabinsk region. URL: <https://rpubs.com/abotalebmostafa/710535> (accessed: 09.01.2021).
2. Abotaleb M.S.A., Makarovskikh T.A. New package “Epidemic.ta” for forecasting Covid-19 infection cases apply Example For forecasting infection cases in the USA. URL: <https://rpubs.com/abotalebmostafa/711262> (accessed: 09.01.2021).
3. Abotaleb M.S.A., Makarovskikh T.A. Research opportunities automatic parameter selection ARIMA models for short term forecast cases Covid-19. URL: <https://rpubs.com/abotalebmostafa/710470> (accessed: 09.01.2021).
4. Abotaleb M.S.A., Makarovskikh T.A. Using auto ARIMA for forecasting Deaths of COVID-19

- in Chelyabinsk. URL: <https://rpubs.com/abotalebmostafa/710528> (accessed: 09.01.2021).
5. Coronavirus: statistics. URL: <https://yandex.ru/covid19/stat> (accessed: 09.01.2021).
 6. Lakman I.A., Agapitov A.A., Sadikova L.F. et al. Possibilities of mathematical forecasting of coronavirus infection in the Russian Federation. *Arterialnaya gipertenzia*. 2020. Vol. 26, no. 3. P. 288–294. (in Russian) DOI: 10.18705/1607-419X-2020-26-3-288-294.
 7. Everitt B.S. *The Cambridge Dictionary of Statistics*. The Cambridge Dictionary of Statistics. Ed. 4. 2010. 736 p.
 8. Abotaleb M.S.A. Predicting COVID-19 Cases using Some Statistical Models: An Application to the Cases Reported in China Italy and USA. *Academic Journal of Applied Mathematical Sciences*. 2020. Vol. 6, no. 4. P. 32–40. DOI: 10.32861/ajams.64.32.40.
 9. Chaurasia V., Pal S. COVID-19 Pandemic: ARIMA and Regression Model-Based Worldwide Death Cases Predictions. *SN Computer Science*, 2020. Art. 1:288. DOI: 10.1007/s42979-020-00298-6.
 10. COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. URL: <https://github.com/CSSEGISandData/COVID-19> (accessed: 23.11.2020).
 11. COVID-19: Status of Cases in Toronto. URL: <https://www.toronto.ca/home/covid-19/covid-19-latest-city-of-toronto-news/covid-19-status-of-cases-in-toronto/> (accessed: 23.11.2020).
 12. Elie R., Hubert E., Turinici G. Contact rate epidemic control of COVID-19: an equilibrium view. *Mathematical Modelling of Natural Phenomena*. 2020. Vol. 15, art. 35. DOI: 10.1051/mmnp/2020022.
 13. Hernandez-Matamorosb A., Fujitaa H., Hayashib T., Perez-Meana H. Forecasting of COVID-19 per regions using ARIMA models and polynomial functions. *Applied Soft Computing Journal*. 2020. Vol. 96, art. 106610. DOI: 10.1016/j.asoc.2020.106610.
 14. Hussain N., Li B. Using R-studio to examine the COVID-19 Patients in Pakistan Implementation of SIR Model on Cases. *International Journal of Scientific Research in Multidisciplinary Studies*. 2020. Vol. 6, no. 8. P. 54–59. DOI: 10.13140/RG.2.2.32580.04482.
 15. Hyndman R.J., Athanasopoulos G. *Forecasting: principles and practice*. OTexts, 2018. 384 p.
 16. Kumar M., Gupta S., Kumar K., Sachdeva M. Spreading of Covid-19 in India, Italy, Japan, Spain, UK, US: A Prediction Using ARIMA and LSTM Model. *Digital Government: Research and Practice*. 2020. Vol. 1, no. 4, art. 24. DOI: 10.1145/3411760.
 17. Panda M. Application of ARIMA and Holt-Winters forecasting model to predict the spreading of COVID-19 for India and its states. DOI: 10.1101/2020.07.14.20153908 (accessed: 10.02.2021).
 18. Perone G. ARIMA forecasting of COVID-19 incidence in Italy, Russia, and the USA. DOI: 10.2139/ssrn.3612402.
 19. Shokeralla A.A.A., Sameeh F.R.I., Musa A.G.M., Zahrani S. Prediction the daily number of confirmed cases of Covid-19 in Sudan with ARIMA and Holt-Winter exponential smoothing. *International Journal of Development Research*. 2020. Vol. 10, no. 8. P. 39408–39413 DOI: 10.37118/ijdr.19811.08.2020.

20. Sun D., Duan L., Xiong J., Wang D. Modelling and forecasting the spread tendency of the COVID-19 in China. BMC Infectious Diseases. DOI: 10.21203/rs.3.rs-26772/v1.
21. Volpert V., Banerjee M., d'Onofrio A., Lipniacki T., Petrovskii S., Tran V.Ch. Coronavirus: scientific insights and societal aspects. Mathematical Modelling of Natural Phenomena. 2020. Vol. 15, art. E2. DOI: 10.1051/mmnp/2020010.
22. Volpert V., Banerjee M., Petrovskii S., Volpert V. On a quarantine model of coronavirus infection and data analysis. Mathematical Modelling of Natural Phenomena. 2020. Vol. 15, art. 24. DOI: 10.1051/mmnp/2020006.

О ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ*

© 2021 Л.Б. Соколинский, И.М. Соколинская

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: leonid.sokolinsky@susu.ru, irina.sokolinskaya@susu.ru

Поступила в редакцию: 02.04.2021

В статье рассматривается масштабируемый алгоритм FRaGenLP для генерации больших совместных случайных задач линейного программирования произвольной размерности n на кластерных вычислительных системах. Для обеспечения совместности и ограниченности допустимой области система ограничений включает в себя $2n+1$ стандартных неравенств, называемых опорными. Случайные неравенства добавляются в систему последовательно так, чтобы сохранялась совместность ограничений. Кроме этого, вводятся две метрики «похожести», которые препятствуют добавлению нового случайного неравенства, «похожего» на какое-либо из уже включенных в систему, включая опорные. Также отклоняются случайные неравенства, которые при фиксированной целевой функции не влияют на решение опорной задачи линейного программирования. Параллельная реализация алгоритма FRaGenLP выполнена на языке C++ с использованием параллельного BSF-каркаса, инкапсулирующего в проблемно-независимой части своего кода все аспекты, связанные с распараллеливанием программы на базе библиотеки MPI. Приводятся результаты масштабных вычислительных экспериментов на кластерной вычислительной системе, подтверждающие эффективность использованного подхода.

Ключевые слова: случайная задача линейного программирования, генератор задач, FRaGenLP, кластерные вычислительные системы, BSF-каркас.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Соколинский Л.Б., Соколинская И.М. О генерации случайных задач линейного программирования на кластерных вычислительных системах // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 38–52. DOI: 10.14529/cmse210203.

Введение

Эпоха больших данных [1, 2] породила задачи линейного программирования (ЛП) сверхбольших размерностей [3]. Подобные задачи возникают в экономике, индустрии, логистике, статистике, квантовой физике и других областях. Решение таких сверхбольших задач невозможно без масштабируемых параллельных алгоритмов, ориентированных на кластерные вычислительные системы. В соответствии с этим в последние годы интенсифицировались усилия по разработке новых и модернизации известных параллельных алгоритмов решения задач ЛП. В качестве примеров можно привести работы [4–8]. При разработке новых масштабируемых алгоритмов для решения сверхбольших задач линейного программирования возникает необходимость их тестирования на известных и случайных задачах. Одним из наиболее известных репозиториях больших задач линейного программирования является Netlib-Lp [9]. Данный репозиторий считается эталонным при апробации новых алгоритмов решения задач линейного программирования. Однако при

* Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2021»

отладке решателей часто возникает необходимость в генерации случайных задач ЛП с определенными характеристиками, из которых основными являются размерность пространства и количество ограничений. В работе [10] был предложен один из первых методов генерации случайных задач ЛП с известными решениями. Метод позволяет генерировать тестовые задачи произвольного размера с широким диапазоном числовых характеристик. Идея метода заключается в том, что за основу берется задача ЛП с известным решением, которая затем модифицируется случайным образом так, чтобы решение не изменилось. Основным недостатком этого метода является то, что предварительная фиксация оптимального решения в значительной мере ограничивает случайный характер итоговой задачи ЛП. В [11] предложен генератор GENGUB, позволяющий строить случайные задачи ЛП с известным решением и заданными характеристиками, такими как размер задачи, плотность ненулевых значений матрицы коэффициентов, степень вырожденности, количество избыточных неравенств и др. Отличительной особенностью генератора GENGUB является введение *обобщенных верхних границ*, представляющих собой совокупность ограничений, в которых каждая переменная появляется как минимум один раз (имеет ненулевой коэффициент). Указанный метод имеет тот же недостаток, что и предыдущий: предварительная фиксация оптимального решения в значительной мере ограничивает случайный характер результирующей задачи ЛП. В статье [12] предложен метод генерации случайных задач ЛП с заранее выбранным типом решения: ограниченное или неограниченное, единственное или множественное. В зависимости от выбранного типа решения конструируется допустимая область задачи ЛП как объединение линейного многообразия, конуса и многогранника. Каждая из перечисленных структур генерируется с помощью случайных векторов с целочисленными координатами. Далее генерируется целевая функция, приводящая к решению требуемого типа. Описанный генератор задач ЛП используется в основном для учебных целей и мало подходит для тестирования новых алгоритмов линейного программирования в силу ограниченности многообразия генерируемых задач.

В данной статье предлагается альтернативный способ генерации случайных задач ЛП, особенность которого состоит в том, что генерируются совместные задачи заданной размерности с неизвестным решением. Такая задача подается на вход решателя. Полученное решение проверяется программой-валидатором [13], которая сертифицирует решение. Предложенный метод генерации случайных задач ЛП реализован в виде параллельной программы *FRaGenLP* (*Feasible Random Generator of LP*) для кластерных вычислительных систем. Статья организована следующим образом. В разделе 1 дается формальное описание предлагаемого метода генерации случайных задач ЛП и приводится последовательная версия алгоритма FRaGenLP. В разделе 2 рассматривается параллельная версия алгоритма FRaGenLP. В разделе 3 предлагается ее реализация с использованием параллельного BSF-каркаса и приводятся результаты масштабных вычислительных экспериментов на кластерной вычислительной системе, подтверждающие эффективность предложенного подхода. В заключении суммируются полученные результаты и раскрываются планы по использованию генератора FRaGenLP для разработки искусственной нейронной сети, способной решать задачи ЛП большой размерности.

1. Метод генерации случайных задач ЛП

Метод генерации случайных задач ЛП, используемый в программе FRaGenLP, позволяет генерировать случайные совместные ограниченные задачи ЛП произвольной размерности n , решение которых заранее неизвестно. Для обеспечения корректности задачи ЛП система ограничений включает в себя следующие *опорные неравенства*:

$$\left\{ \begin{array}{ll} x_1 & \leq \alpha \\ & x_2 & \leq \alpha \\ & & \ddots & \dots \dots \\ & & & x_n & \leq \alpha \\ -x_1 & & & & \leq 0 \\ & -x_2 & & & \leq 0 \\ & & \ddots & & \dots \dots \\ & & & -x_n & \leq 0 \\ x_1 & +x_2 & \dots & +x_n & \leq \alpha(n-1)+100 \end{array} \right. \quad (1)$$

Здесь $\alpha \in \mathbb{R}_{>0}$ — положительный масштабирующий коэффициент, являющийся параметром программы FRaGenLP. Количество опорных неравенств равно $2n+1$. Количество случайных неравенств определяется параметром $d \in \mathbb{Z}_{\geq 0}$. Общее количество неравенств m определяется по формуле

$$m = 2n + 1 + d. \quad (2)$$

Коэффициенты целевой функции задаются вектором

$$c = \theta(n, n-1, n-2, \dots, 1), \quad (3)$$

где $\theta \in \mathbb{R}_{>0}$ — положительный постоянный множитель, являющийся параметром программы FRaGenLP и удовлетворяющий условию

$$\theta \leq \frac{\alpha}{2}. \quad (4)$$

Везде далее будем предполагать, что необходимо найти допустимую точку, в которой достигается максимум целевой функции. При количестве случайных неравенств $d=0$ генерируется задача ЛП, ограничения которой включают только опорные неравенства (1). В этом случае задача ЛП имеет следующее единственное решение:

$$\bar{x} = (\alpha, \dots, \alpha, \alpha/2). \quad (5)$$

Коэффициенты случайного неравенства $a_i = (a_{i_1}, \dots, a_{i_n})$ и правая часть b_i вычисляются с помощью функции $\text{rand}(l, r)$, генерирующей случайное вещественное число из интервала $[l, r]$ ($l, r \in \mathbb{R}; l < r$), и функции $\text{rsgn}()$, случайным образом выбирающей число из множества $\{1, -1\}$:

$$\begin{aligned} a_{ij} &:= \text{rsgn}() \cdot \text{rand}(0, a_{max}), \\ b_i &:= \text{rsgn}() \cdot \text{rand}(0, b_{max}). \end{aligned} \quad (6)$$

Здесь $a_{max}, b_{max} \in \mathbb{R}_{>0}$ — константы, являющиеся параметрами программы FRaGenLP. В качестве знака неравенства всегда выбирается \leq . Введем следующие обозначения:

$$f(x) = \langle c, x \rangle; \quad (7)$$

$$h_{ctr} = (\alpha/2, \dots, \alpha/2); \quad (8)$$

$$\text{dist}(a_i, b_i, h_{ctr}) = \frac{|\langle a_i, h_{ctr} \rangle - b_i|}{\|a_i\|}; \quad (9)$$

$$\pi(x, a_i, b_i) = x - \frac{\langle a_i, x \rangle - b_i}{\|a_i\|^2} a_i. \quad (10)$$

Формула (7) определяет целевую функцию задачи ЛП. Здесь и далее $\langle \bullet, \bullet \rangle$ обозначает скалярное произведение векторов. Формула (8) определяет центральную точку *ограничивающего гиперкуба*, задаваемого первыми $2n$ неравенствами системы (1). Формула (9) определяет функцию $\text{dist}(a_i, b_i, h_{ctr})$, вычисляющую расстояние от гиперплоскости $\langle a_i, x \rangle = b_i$ до центра ограничивающего гиперкуба. Здесь и далее $\|\bullet\|$ обозначает евклидову норму. Формула (10) определяет вектор-функцию, вычисляющую ортогональную проекцию точки x на гиперплоскость $\langle a_i, x \rangle = b_i$.

Случайное неравенство $\langle a_i, x \rangle \leq b_i$ получается путем генерации координат вектора a_i и свободного члена b_i с помощью датчика псевдослучайных дробно-рациональных чисел. Сгенерированное случайное неравенство добавляется в систему ограничений в том, и только в том случае, когда выполняются следующие условия:

$$\langle a_i, h_{ctr} \rangle \leq b_i; \quad (11)$$

$$\rho < \text{dist}(a_i, b_i, h_{ctr}) \leq \theta; \quad (12)$$

$$f(\pi(h_{ctr}, a_i, b_i)) > f(h_{ctr}); \quad (13)$$

$$\forall l \in \{1, \dots, i-1\} : \neg \text{like}(a_i, b_i, a_l, b_l). \quad (14)$$

Условие (11) требует, чтобы центр ограничивающего гиперкуба являлся допустимой точкой для добавляемого неравенства. Если это условие не выполняется, то вместо неравенства $\langle a_i, x \rangle \leq b_i$ можно взять неравенство $-\langle a_i, x \rangle \leq -b_i$. Условие (12) требует, чтобы расстояние от гиперплоскости $\langle a_i, x \rangle = b_i$ до центра h_{ctr} ограничивающего гиперкуба было больше ρ и меньше θ . Константа $\rho \in \mathbb{R}_{>0}$ является параметром программы FRaGenLP и должна удовлетворять условию $\rho < \theta$, где θ , в свою очередь, удовлетворяет условию (4). Условие (13) требует, чтобы значение целевой функции в точке проекции h_{ctr} на гиперплоскость $\langle a_i, x \rangle = b_i$ было больше, чем в самой точке h_{ctr} . Это условие в сочетании с (11) и (12) отсекает ограничения, которые не могут повлиять на решение задачи ЛП. Условие (14) требует, чтобы новое неравенство было *непохоже* на все ранее добавленные, включая опорные. Указанное условие использует булевскую функцию *like*, определяющую *похожесть (likeness)* двух неравенств $\langle a_i, x \rangle \leq b_i$ и $\langle a_l, x \rangle \leq b_l$ по следующей формуле:

$$\text{like}(a_i, b_i, a_l, b_l) = \left\| \frac{a_i}{\|a_i\|} - \frac{a_l}{\|a_l\|} \right\| < L_{max} \wedge \left| \frac{b_i}{\|a_i\|} - \frac{b_l}{\|a_l\|} \right| < S_{min}. \quad (15)$$

Константы $L_{max}, S_{min} \in \mathbb{R}_{>0}$ являются параметрами программы FRaGenLP. При этом параметр L_{max} должен удовлетворять условию

$$L_{max} \leq 0.7 \quad (16)$$

(обоснование этому ограничению будет дано ниже). В соответствии с формулой (15) неравенства $\langle a_i, x \rangle \leq b_i$ и $\langle a_l, x \rangle \leq b_l$ считаются похожими, если выполняются следующие два условия:

$$\left\| \frac{a_i}{\|a_i\|} - \frac{a_l}{\|a_l\|} \right\| < L_{max}; \quad (17)$$

$$\left| \frac{b_i}{\|a_i\|} - \frac{b_l}{\|a_l\|} \right| < S_{min}. \quad (18)$$

Условие (17) оценивает параллельность гиперплоскостей $\langle a_i, x \rangle = b_i$ и $\langle a_l, x \rangle = b_l$, ограничивающих допустимые области исходных неравенств. Для этого сначала вычисляются единичные векторы нормалей $e_i = a_i / \|a_i\|$ и $e_l = a_l / \|a_l\|$ к гиперплоскостям $\langle a_i, x \rangle = b_i$ и $\langle a_l, x \rangle = b_l$. Затем вычисляется норма разности единичных векторов нормалей: $\delta = \|e_i - e_l\|$. Если $\delta = 0$, то гиперплоскости параллельны. Если $0 < \delta < L_{max}$, то гиперплоскости считаются *почти параллельными*. Условие (18) оценивает «близость» гиперплоскостей $\langle a_i, x \rangle = b_i$ и $\langle a_l, x \rangle = b_l$ относительно друг друга. Для этого сначала вычисляются нормализованные свободные члены $\beta_i = b_i / \|a_i\|$ и $\beta_l = b_l / \|a_l\|$. Затем вычисляется модуль их разности $\sigma = |\beta_i - \beta_l|$. Если гиперплоскости параллельны и $\sigma = 0$, то эти гиперплоскости совпадают. Если $0 < \sigma < S_{min}$, то гиперплоскости считаются *почти близкими*. Два линейных неравенства в \mathbb{R}^n считаются *похожими*, если соответствующие им гиперплоскости являются почти параллельными и почти близкими.

Ограничение (16) для параметра L_{max} основано на следующем утверждении.

Утверждение 1. Пусть заданы два единичных вектора $e, e' \in \mathbb{R}^n$ и угол $\varphi < \pi$ между ними. Тогда

$$\|e - e'\| = \sqrt{2(1 - \cos \varphi)}. \quad (19)$$

Доказательство. По определению нормы в евклидовом пространстве имеем

$$\|e - e'\|^2 = \sum_j (e_j - e'_j)^2 = \sum_j (e_j^2 - 2e_j e'_j + e_j'^2) = \sum_j e_j^2 - 2 \sum_j e_j e'_j + \sum_j e_j'^2 = \sqrt{1 - 2\langle e_j, e'_j \rangle} + 1.$$

Таким образом

$$\|e - e'\| = \sqrt{2(1 - \langle e_j, e'_j \rangle)}. \quad (20)$$

По определению угла в евклидовом пространстве для единичных векторов имеем

$$\langle e_j, e'_j \rangle = \cos \varphi.$$

Подставив правую часть в (20), получаем

$$\|e - e'\| = \sqrt{2(1 - \cos \varphi)}.$$

Утверждение доказано.

Разумно считать два единичных вектора e, e' почти параллельными, если угол φ между ними меньше, чем $\pi/4$. В этом случае согласно (19) имеем

$$\|e - e'\| < \sqrt{2\left(1 - \cos\frac{\pi}{4}\right)}.$$

Учитывая, что $\cos(\pi/4) \approx 0.707$, получаем требуемую оценку:

$$\|e - e'\| < 0.7.$$

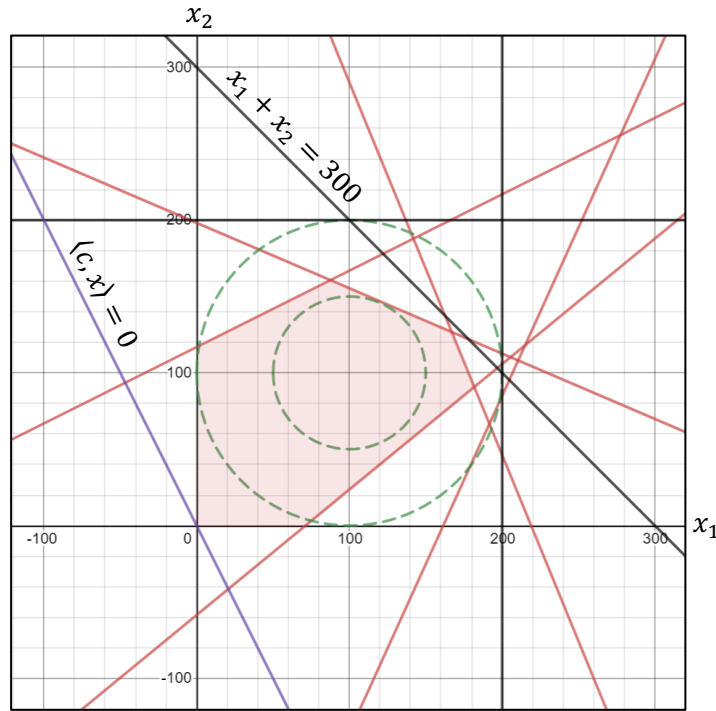


Рис. 1. Случайная задача линейного программирования для $n = 2, d = 5, \alpha = 200, \theta = 100, \rho = 50, S_{min} = 100, L_{max} = 0.35, a_{max} = 1000, b_{max} = 10000$ (нарисовано с помощью графического калькулятора Desmos [14])

На рис. 1 представлена случайная задача, сгенерированная программой FRaGenLP в соответствии с выше описанным методом для размерности $n = 2$ с использованием следующих значений параметров: $d = 5, \alpha = 200, \theta = 100, \rho = 50, S_{min} = 100, L_{max} = 0.35, a_{max} = 1000, b_{max} = 10000$. Фиолетовым цветом на рисунке обозначена прямая, определяемая коэффициентами целевой функции; черными линиями обозначены прямые, соответствующие опорным неравенствам; красными линиями обозначены прямые, соответствующие случайным неравенствам. Для наглядности на рисунке также обозначены зелеными пунктирными линиями большая и малая окружности, задаваемые уравнениями $(x_1 - 100)^2 + (x_2 - 100)^2 = 100^2$ и $(x_1 - 100)^2 + (x_2 - 100)^2 = 50^2$ соответственно. Любая случайная прямая должна пересекать большую окружность и не пересекать малую окружность в соответствии с условием (12). Полупрозрачным красным цветом выделена область допустимых точек получившейся задачи линейного программирования.

Алгоритм 1. Последовательный алгоритм генерации случайной задачи ЛП

```

1: input  $n, d, \alpha, \theta, \rho, S_{min}, L_{max}, a_{max}, b_{max}$ 
2:  $k := 0$ 
3:  $A := []$ 
4:  $B := []$ 
5:  $AddSupport(A, B)$ 
6: for  $j = n \dots 1$  do  $c_j := \theta \cdot j$ 
7: if  $d = 0$  goto 20
8: for  $j = 1 \dots n$  do  $a_j := rsign() \cdot rand(0, a_{max})$ 
9:  $b := rsign() \cdot rand(0, b_{max})$ 
10: if  $\langle a, h_{ctr} \rangle \leq b$  goto 13
11: for  $j = 1 \dots n$  do  $a_j := -a_j$ 
12:  $b := -b$ 
13: if  $dist(a, b, h_{ctr}) < \rho$  or  $dist(a, b, h_{ctr}) > \theta$  goto 8
14: if  $f(\pi(h_{ctr}, a_i, b_i)) > f(h_{ctr})$  goto 8
15: for all  $(\bar{a}, \bar{b}) \in (A, B)$  do if  $like(a, b, \bar{a}, \bar{b})$  goto 8
16:  $A := A \# a$ 
17:  $B := B \# b$ 
18:  $k := k + 1$ 
19: if  $k < d$  goto 8
20: output  $A, B, c$ 
21: stop

```

Описанный метод представлен в виде последовательного алгоритма 1. На шаге 1 вводятся параметры алгоритма. На шаге 2 счетчику случайных неравенств присваивается значение 0. На шаге 3 формируется пустой список строк матрицы A . На шаге 4 формируется пустой список строк свободных членов B . На шаге 5 в списки A и B добавляются коэффициенты и свободные члены опорных неравенств (1). На шаге 6 проверяется количество случайных неравенств d : если $d = 0$, то происходит вывод только опорных неравенств, после чего алгоритм заканчивает свою работу. На шаге 7 генерируются коэффициенты целевой функции в соответствии с формулой (3). На шагах 8 и 9 генерируются коэффициенты и свободный член нового случайного неравенства. На шаге 10 проверяется условие (11). Если это условие не верно, то знаки коэффициентов и свободного члена меняются на противоположные (шаги 11, 12). На шаге 13 проверяется условие (12). На шаге 14 проверяется условие (13). На шаге 15 проверяется условие (14). На шаге 16 коэффициенты нового случайного неравенства добавляются в матрицу A . На шаге 17 свободный член нового случайного неравенства добавляется в столбец B . На шаге 18 счетчик неравенств увеличивается на единицу. На шаге 19 проверяется, достигло ли количество неравенств требуемого числа, и, при необходимости, осуществляется переход к следующей итерации. На шаге 20 происходит вывод результатов. Шаг 21 завершает работу алгоритма.

2. Параллельный алгоритм генерации случайных задач ЛП

Неявные циклы с возвратом на метку 8, возникающие на шагах 13–15 алгоритма 1, могут потребовать значительных временных затрат. Так, при генерации задачи ЛП, представленной на рис. 1, возвраты на метку 8 осуществлялись 112581 раз с шага 13, 32771 раз с шага 14, и 726 раз с шага 15. В силу этого генерация большой случайной задачи ЛП на обычном персональном компьютере может длиться несколько суток. Поэтому нами

Алгоритм 2. Параллельный алгоритм генерации случайной задачи ЛП

Мастер	l -й рабочий ($l=1, \dots, L$)
M1: input $n, d, \alpha, \theta, \rho, S_{min}, L_{max}, a_{max}, b_{max}$	W1: input $n, d, \alpha, \theta, \rho, S_{min}, L_{max}, a_{max}, b_{max}$
M2: $k := 0$	W2: if $d \leq 0$ goto W37
M3: $A_S := []$	W3: $A_S := []$
M4: $B_S := []$	W4: $B_S := []$
M5: <i>AddSupport</i> (A_S, B_S)	W5: <i>AddSupport</i> (A_S, B_S)
M6: for $j = n \dots 1$ do $c_j := \theta \cdot j$	W6: for $j = 1 \dots n$ do begin
M7: $A_R := []$	W7: $a_j^{(l)} := rsign() \cdot rand(0, a_{max})$
M8: $B_R := []$	W8: end
M9: if $d > 0$ goto M19	W9: $b^{(l)} := rsign() \cdot rand(0, b_{max})$
M10: output A_S, B_S, c	W10: if $\langle a^{(l)}, h_{cntr} \rangle \leq b^{(l)}$ goto W13
M11: goto M37	W11: for $j = 1 \dots n$ do $a_j := -a_j$
M12:	W12: $b^{(l)} := -b^{(l)}$
M13:	W13: if $dist(a^{(l)}, b^{(l)}, h_{cntr}) < \rho$ goto W6
M14:	W14: if $dist(a^{(l)}, b^{(l)}, h_{cntr}) > \theta$ goto W6
M15:	W15: if $f(\pi(h_{cntr}, a^{(l)}, b^{(l)})) > f(h_{cntr})$ goto W6
M16:	W16: for all $(\bar{a}, \bar{b}) \in (A_S, B_S)$ do begin
M17:	W17: if $like(a^{(l)}, b^{(l)}, \bar{a}, \bar{b})$ goto W6
M18:	W18: end
M19: <i>RecvFromAllWorkers</i> ($a^{(1)}, b^{(1)}, \dots, a^{(L)}, b^{(L)}$)	W19: <i>SendToMaster</i> ($a^{(l)}, b^{(l)}$)
M20: for $l = 1 \dots L$ do begin	W20:
M21: $is_like := false$	W21:
M22: for all $(\bar{a}, \bar{b}) \in (A_R, B_R)$ do begin	W22:
M23: if $like(a^{(l)}, b^{(l)}, \bar{a}, \bar{b})$ then begin	W23:
M24: $is_like := true$	W24:
M25: goto M28	W25:
M26: end	W26:
M27: end	W27:
M28: if is_like continue	W28:
M29: $A_R := A_R \# a^{(l)}$	W29:
M30: $B_R := B_R \# b^{(l)}$	W30:
M31: $k := k + 1$	W31:
M32: if $k = d$ goto M34	W32:
M33: end	W33:
M34: <i>SendToAllWorkers</i> (k)	W34: <i>RecvFromMaster</i> (k)
M35: if $k < d$ goto M19	W35: if $k < d$ goto W6
M36: output A_S, B_S, A_R, B_R, c	W36:
M37: stop	W37: stop

была разработана параллельная версия генератора задач FRaGenLP для кластерных вычислительных систем. Эта версия представлена в виде алгоритма 2. Данный алгоритм разработан в соответствии с моделью параллельных вычислений BSF [15], использующей парадигму «мастер-рабочий» [16]. В соответствии с этой моделью узел-мастер является центром управления и коммуникации. Все узлы-рабочие выполняют один и тот же код, но над разными данными.

Сначала рассмотрим *шаги, выполняемые узлом-мастером*. На шаге M1 мастер считывает параметры задачи. На шаге M2 счетчику случайных неравенств присваивается значение 0. В параллельной версии опорные и случайные неравенства хранятся в разных списках. На шагах M3 и M4 мастер создает пустые списки A_S и B_S для коэффициентов и свободных членов опорных неравенств. На шаге M5 в эти списки добавляются опорные

неравенства (1). На шаге M6 генерируются коэффициенты целевой функции в соответствии с формулой (3). На шагах M7 и M8 создаются пустые списки A_R и B_R для коэффициентов и свободных членов случайных неравенств. На шаге M9 проверяется количество случайных неравенств d : если $d \leq 0$, то происходит вывод только опорных неравенств (шаг M10), после чего мастер заканчивает свою работу (переход с шага M11 на шаг M37). Если $d > 0$, то мастер продолжает свою работу. На шаге M19 мастер получает по одному случайному неравенству от всех рабочих. Рабочие обеспечивают для своих неравенств выполнение свойств (11)–(13) и проверяют их похожесть на все опорные неравенства. Мастер в цикле M20–M33 проверяет все поступившие случайные неравенства на похожесть с ранее включенными в списки A_R и B_R случайными неравенствами (вложенный цикл M22–M27). Если обнаруживается похожее случайное неравенство (шаги M23–M26), то новое неравенство игнорируется, и осуществляется переход к проверке следующего поступившего неравенства (шаг M28). Если похожих неравенств не нашлось, новое случайное неравенство добавляется мастером в списки A_R , B_R (шаги M29, M30), и счетчик случайных неравенств увеличивается на единицу (шаг M31). Если количество неравенств достигло заданного значения (шаг M32), то происходит выход из внешнего цикла, в противном случае проверки продолжаются. После завершения проверок и добавлений поступивших случайных неравенств мастер рассылает всем рабочим общее количество случайных неравенств, включенных в систему на данный момент (шаг M34). Если это количество не достигло требуемого, то происходит переход к следующей глобальной итерации (шаг M35), в противном случае мастер выводит результаты и завершает свою работу (шаги M36, M37).

Теперь рассмотрим *шаги, выполняемые l-м рабочим*. На шаге W1 рабочий считывает параметры задачи. На шаге W2 проверяется количество случайных неравенств d : если $d \leq 0$, то рабочий немедленно заканчивает свою работу, в противном случае создаются пустые списки A_S и B_S для коэффициентов и свободных членов опорных неравенств (шаги W3, W4). На шаге W5 в эти списки добавляются опорные неравенства (1). На шагах W6–W9 рабочий генерирует новое случайное неравенство. На шаге W10 проверяется условие (11). Если это условие не верно, то знаки коэффициентов и свободного члена сгенерированного неравенства меняются на противоположные (шаги W11, W12). Шаги W13–W15 проверяют условия (12), (13). Шаги W16–W18 проверяют похожесть сгенерированного неравенства на опорные неравенства с использованием формулы (14). Если хотя бы одно из этих условий не выполняется, осуществляется переход на шаг W6 для повторной генерации нового случайного неравенства. Если все указанные условия выполнены, рабочий пересылает мастеру построенное случайное неравенство (шаг W19). После этого рабочий ждет, когда мастер пришлет ему новое значение счетчика случайных неравенств k (шаг W34). Если k меньше заданного количества случайных неравенств d , то происходит переход на шаг W6 для генерации следующего случайного неравенства (шаг W35), в противном случае рабочий завершает свою работу (шаг W37).

3. Программная реализация и вычислительные эксперименты

Параллельный алгоритм 2 был нами реализован на языке C++ с использованием параллельного BSF-каркаса [17, 18]. BSF-каркас базируется на модели параллельных вычислений BSF [15] и инкапсулирует в проблемно независимой части своего кода все аспекты, связанные с распараллеливанием программы с помощью библиотеки MPI [19]. BSF-каркас требует преобразования алгоритма в форму, работающую со списками с использованием функций высшего порядка *Map* и *Reduce*, определяемых формализмом Бирда—Миртенса (Bird—Meertens formalism) [20]. Это преобразование было сделано следующим образом. Длина списков *Map* и *Reduce* устанавливалась равной количеству рабочих MPI-процессов. Элементы списка *Map* определялись как пустые структуры:

```
struct PT_bsf_mapElem_T {}.
```

Каждый элемент списка *Reduce* хранил коэффициенты и свободный член одного случайного неравенства $\langle a, x \rangle \leq b$:

```
struct PT_bsf_reduceElem_T {float a[n]; float b}.
```

Каждый рабочий MPI-процесс генерировал по одному случайному неравенству с помощью функции *PC_bsf_MapF*, которая включала в себя шаги W6–W18 алгоритма 2. Неравенство, удовлетворяющее всем условиям, записывалось рабочим в локальный список *Reduce*, состоящий из одного элемента. Мастер получал от каждого рабочего сгенерированные элементы и помещал их в свой список *Reduce* (это выполнялось проблемно-независимой частью BSF-каркаса). Полученные элементы вручную проверялись на похожесть с ранее добавленными случайными неравенствами и, при отсутствии конфликтов, добавлялись в списки A_R, B_R . Указанная обработка (шаги M20–M33) была нами реализована в предопределенной функции *PC_bsf_ProcessResults*. Исходные коды параллельной программы FRaGenLP свободно доступны в сети Интернет по адресу <https://github.com/leonid-sokolinsky/BSF-LPP-Generator>.

С использованием указанной программы нами были проведены масштабные вычислительные эксперименты на вычислительном кластере «Торнадо ЮУрГУ» [21], характеристики которого приведены в табл. 1. Вычисления проводились для размерности n , равной 3 000, 5 500 и 15 000. Количество неравенств, соответственно, составляло 6 301, 10 001 и 31 501. Из них случайных: 300, 500, 1500 соответственно. Во всех экспериментах были установлены следующие значения параметров, введенных в разделе 1: длина ребра ограничивающего гиперкуба $\alpha = 200$, радиус большой гиперсферы $\theta = 100$, радиус малой гиперсферы $\rho = 50$, верхняя граница «почти параллельности» для гиперплоскостей $L_{max} = 0.35$, минимальная допустимая близость для гиперплоскостей $S_{min} = 100$, максимальное допустимое абсолютное значение при генерации коэффициентов случайных неравенств $a_{max} = 1000$, максимальное допустимое абсолютное значение при генерации правых частей случайных неравенств $b_{max} = 10000$. Результаты экспериментов представлены на рис. 2. Время генерации случайной задачи ЛП с 31501 ограничениями на конфигурации из одного узла-мастера и одного узла-рабочего составило 12 минут. На конфигурации из одного узла-мастера и 170 узлов-рабочих генерация такой же задачи заняла 22 секунды. Анализ результатов показывает, что граница масштабируемости предложенного алгоритма существенно зависит от размерности задачи (под границей масштабируемости здесь понимается максимум кривой ускорения). При $n = 3000$ граница масштабируемости составила приблизительно 50 процессорных узлов. Для задачи размерности $n = 5000$ эта

Таблица 1. Характеристики кластера «Торнадо ЮУрГУ»

Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfiniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

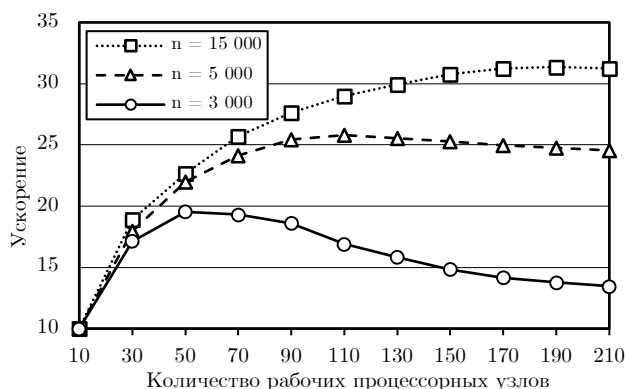


Рис. 2. Графики ускорения алгоритма FRaGenLP для различных размерностей

граница увеличилась до 110 узлов, а на задаче размерности $n = 15000$ она достигла 200 узлов. Дальнейшее увеличение размерности задачи приводило к нехватке оперативной памяти на процессорных узлах. Следует отметить, что граница масштабируемости предложенного алгоритма также существенно зависит от количества случайных неравенств. Увеличение этого числа в 10 раз приводило к двукратному уменьшению границы масштабируемости. Это объясняется тем, что при увеличении количества узлов-рабочих значительно возрастала доля последовательных вычислений, выполняемых узлом-мастером на шагах M20–M27, в течение которых узлы-рабочие простаивали.

Заключение

В данной работе представлен параллельный алгоритм FRaGenLP для генерации случайных совместных ограниченных задач ЛП на кластерных вычислительных системах. Генерируемые системы ограничений наряду со случайными неравенствами включают в себя стандартный набор неравенств, называемых опорными. Опорные неравенства гарантируют ограниченность допустимой области задачи ЛП. С геометрической точки зрения допустимая область опорных неравенств представляет собой гиперкуб, прилежащий к осям координат, у которого срезана дальняя от центра координат вершина. Целевая функция задается определенным образом, так, чтобы ее коэффициенты монотонно убывали. Коэффициенты и свободные члены случайных неравенств генерируются с помощью датчика случайных чисел. Если допустимая область сгенерированного случайного неравенства не включает в себя центр ограничивающего гиперкуба, то знак неравенства меняется на противоположный. Однако не всякое случайное неравенство включается в систему. Отклоняются случайные неравенства, которые не могут повлиять на решение задачи ЛП при заданной целевой функции. Также отклоняются все неравенства, для которых ограничивающая гиперплоскость пересекает малую гипертферу, расположенную в центре ограничивающего гиперкуба (это гарантирует совместность). Кроме того, отклоняется всякое случайное неравенство, «похожее» на хотя бы одно неравенство, уже добавленное в систему (включая опорные). Для определения «похожести» неравенств введены две формальные метрики: почти параллельности и близости соответствующих им гиперплоскостей. Параллельный алгоритм спроектирован на базе модели параллельных вычислений BSF, использующей парадигму «мастер-рабочий». В соответствии с этой моделью узел-мастер является центром управления и коммуникации. Все узлы-рабочие выполняют один и тот же код, но над разными данными. Параллельная реализация выполнена на языке C++ с использованием параллельного BSF-каркаса, инкапсулирующего в

проблемно независимой части своего кода все аспекты, связанные с распараллеливанием программы с помощью библиотеки MPI. Исходные коды разработанной параллельной программы свободно доступны в сети Интернет по адресу <https://github.com/leonid-sokolinsky/BSF-LPP-Generator>. С использованием этой реализации были проведены масштабные вычислительные эксперименты на кластерной вычислительной системе. Проведенные эксперименты показали, что алгоритм FRaGenLP демонстрирует для размерности 15 000 хорошую масштабируемость вплоть до 200 процессорных узлов. Случайная задача ЛП, включающая в себя 31 501 неравенство, была сгенерирована на конфигурации из 171 процессорного узла за 22 секунды. На одном процессорном узле для этого потребовалось 12 минут. Разработанную программу предполагается использовать для генерации 70 000 образцов для обучения искусственной нейронной сети, способной быстро решать большие задачи ЛП.

Исследование выполнено при финансовой поддержке РФФИ (проект № 20-07-00092-а) и Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

Литература

1. Jagadish H.V. et al. Big data and its technical challenges // Communications of the ACM. 2014. Vol. 57, no. 7. P. 86–94. DOI: 10.1145/2611567.
2. Hartung T. Making Big Sense From Big Data // Frontiers in Big Data. 2018. Vol. 1. P. 5. DOI: 10.3389/fdata.2018.00005.
3. Соколинская И.М., Соколинский Л.Б. О решении задачи линейного программирования в эпоху больших данных // Параллельные вычислительные технологии (ПаВТ'2017): Труды XI международной научной конференции (Казань, 3–7 апреля 2017 г.). Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. С. 471–484.
4. Соколинский Л.Б., Соколинская И.М. Исследование масштабируемости апекс-метода для решения сверхбольших задач линейного программирования на кластерных вычислительных системах // Суперкомпьютерные дни в России: Труды международной конференции (Москва, 21–22 сентября 2020 г.). Москва: МАКС Пресс, 2020. С. 49–59.
5. Mamalis B., Pantziou G. Advances in the Parallelization of the Simplex Method // Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science. Cham: Springer, 2015. Vol. 9295. P. 281–307. DOI: 10.1007/978-3-319-24024-4_17.
6. Huangfu Q., Hall J.A.J. Parallelizing the dual revised simplex method // Mathematical Programming Computation. Springer Verlag, 2018. Vol. 10, no. 1. P. 119–142. DOI: 10.1007/s12532-017-0130-5.
7. Tar P., Stágel B., Maros I. Parallel search paths for the simplex algorithm // Central European Journal of Operations Research. Springer Verlag, 2017. Vol. 25, no. 4. P. 967–984. DOI: 10.1007/s10100-016-0452-9.
8. Yang L., Li T., Li J. Parallel predictor-corrector interior-point algorithm of structured optimization problems // 3rd International Conference on Genetic and Evolutionary Computing, WGEN 2009. 2009. P. 256–259. DOI: 10.1109/WGEN.2009.68.
9. Gay D.M. Electronic mail distribution of linear programming test problems // Mathematical Programming Society COAL Bulletin. 1985. no. 13. P. 10–12.
10. Charnes A. et al. On Generation of Test Problems for Linear Programming Codes // Communications of the ACM. 1974. Vol. 17, no. 10. P. 583–586. DOI: 10.1145/355620.361173.

11. Arthur J.L., Friendewey J.O. GENGUB: A generator for linear programs with generalized upper bound constraints // *Computers and Operations Research*. Pergamon, 1993. Vol. 20, no. 6. P. 565–573. DOI: 10.1016/0305-0548(93)90112-V.
12. Castillo E., Pruneda R.E., Esquivel Mó. Automatic generation of linear programming problems for computer aided instruction // *International Journal of Mathematical Education in Science and Technology*. Taylor & Francis Group, 2001. Vol. 32, no. 2. P. 209–232. DOI: 10.1080/00207390010010845.
13. Dhiflaoui M. et al. Certifying and Repairing Solutions to Large LPs How Good are LP-solvers? // *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. USA: Society for Industrial and Applied Mathematics, 2003. P. 255–256.
14. Desmos Graphing Calculator. URL: <https://www.desmos.com/calculator> (дата обращения: 09.03.2021).
15. Sokolinsky L.B. BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems // *Journal of Parallel and Distributed Computing*. 2021. Vol. 149. P. 193–206. DOI: 10.1016/j.jpdc.2020.12.009.
16. Sahni S., Vairaktarakis G. The master-slave paradigm in parallel computer and industrial settings // *Journal of Global Optimization*. 1996. Vol. 9, no. 3–4. P. 357–377. DOI:10.1007/BF00121679.
17. Sokolinsky L.B. BSF-skeleton. User manual: arXiv:2008.12256 [cs.DC]. 2020. 22 p.
18. Sokolinsky L.B. BSF-skeleton. 2019. URL: <https://github.com/leonid-sokolinsky/BSF-skeleton> (дата обращения: 09.03.2021).
19. Gropp W. MPI 3 and Beyond: Why MPI Is Successful and What Challenges It Faces // *Recent Advances in the Message Passing Interface. EuroMPI 2012. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2012. Vol. 7490. P. 1–9. DOI: 10.1007/978-3-642-33518-1_1.
20. Bird R.S. *Lectures on Constructive Functional Programming // Constructive Methods in Computing Science. NATO ASI Series F: Computer and Systems Sciences*. Berlin, Heidelberg: Springer, 1988. Vol. 55. P. 151–216.
21. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources // *Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016)*. CEUR Workshop Proceedings. Vol. 1576. 2016. P. 561–573.

Соколинский Леонид Борисович, д.ф.-м.н., профессор, проректор по информатизации, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Соколинская Ирина Михайловна, к.ф.-м.н., доцент, кафедра вычислительной математики и высокопроизводительных вычислений, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

ON GENERATOR OF RANDOM PROBLEMS FOR LINEAR PROGRAMMING ON CLUSTER COMPUTING SYSTEMS

© 2021 L.B. Sokolinsky, I.M. Sokolinskaya

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: leonid.sokolinsky@susu.ru, irina.sokolinskaya@susu.ru

Received: 02.04.2021

The article presents and evaluates a scalable FRaGenLP algorithm for generating random linear programming problems of large dimension n on cluster computing systems. To ensure the consistency of the problem and the boundedness of the feasible region, the constraint system includes $2n+1$ standard inequalities, called support inequalities. New random inequalities are generated and added to the system in a manner that ensures the consistency of the constraints. Furthermore, the algorithm uses two likeness metrics to prevent the addition of a new random inequality that is similar to one already present in the constraint system. The algorithm also rejects random inequalities that cannot affect the solution of the linear programming problem bounded by the support inequalities. The parallel implementation of the FRaGenLP algorithm is performed in C++ through the parallel BSF-skeleton, which encapsulates all aspects related to the MPI-based parallelization of the program. We provide the results of large-scale computational experiments on a cluster computing system to study the scalability of the FRaGenLP algorithm.

Keywords: random linear programming problem, problem generator, FRaGenLP, cluster computing systems, BSF-skeleton.

FOR CITATION

Sokolinsky L.B., Sokolinskaya I.M. On Generator of Random Problems for Linear Programming on Cluster Computing Systems. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 38–52. (in Russian) DOI: 10.14529/cmse210203.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Jagadish H.V. et al. Big data and its technical challenges. *Communications of the ACM*. 2014. Vol. 57, no. 7. P. 86–94. DOI: 10.1145/2611567.
2. Hartung T. Making Big Sense From Big Data. *Frontiers in Big Data*. 2018. Vol. 1. P. 5. DOI: 10.3389/fdata.2018.00005.
3. Sokolinskaya I.M., Sokolinsky L.B. On solving the linear programming problem in the era of big data. *Parallel Computational Technologies (PCT'2017): Proceedings of the 11th International Scientific Conference (Kazan, April, 3–7, 2017). Short papers and posters*. Chelyabinsk: SUSU Publishing Center, 2017. P. 471–484. (in Russian)
4. Sokolinsky L.B., Sokolinskaya I.M. Study of the scalability of the apex method for solving ultra-large linear programming problems on cluster computing systems. *Russian Supercomputer Days: Proceedings of the Conference (Moscow, September, 21–22, 2020)*. Moscow, MAKS Press, 2020. P. 49–59. (in Russian)
5. Mamalis B., Pantziou G. *Advances in the Parallelization of the Simplex Method. Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science*. Cham: Springer, 2015. Vol. 9295. P. 281–307. DOI: 10.1007/978-3-319-24024-4_17.

6. Huangfu Q., Hall J.A.J. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*. Springer Verlag, 2018. Vol. 10, no. 1. P. 119–142. DOI: 10.1007/s12532-017-0130-5.
7. Tar P., Stágel B., Maros I. Parallel search paths for the simplex algorithm. *Central European Journal of Operations Research*. Springer Verlag, 2017. Vol. 25, no. 4. P. 967–984. DOI: 10.1007/s10100-016-0452-9.
8. Yang L., Li T., Li J. Parallel predictor-corrector interior-point algorithm of structured optimization problems. 3rd International Conference on Genetic and Evolutionary Computing, WGEC 2009. 2009. P. 256–259. DOI: 10.1109/WGEC.2009.68.
9. Gay D.M. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society COAL Bulletin*. 1985. no. 13. P. 10–12.
10. Charnes A. et al. On Generation of Test Problems for Linear Programming Codes. *Communications of the ACM*. 1974. Vol. 17, no. 10. P. 583–586. DOI: 10.1145/355620.361173.
11. Arthur J.L., Friendewey J.O. GENGUB: A generator for linear programs with generalized upper bound constraints. *Computers and Operations Research*. Pergamon, 1993. Vol. 20, no. 6. P. 565–573. DOI: 10.1016/0305-0548(93)90112-V.
12. Castillo E., Pruneda R.E., Esquivel Mó. Automatic generation of linear programming problems for computer aided instruction. *International Journal of Mathematical Education in Science and Technology*. Taylor & Francis Group, 2001. Vol. 32, no. 2. P. 209–232. DOI: 10.1080/00207390010010845.
13. Dhiflaoui M. et al. Certifying and Repairing Solutions to Large LPs How Good are LP-solvers? SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. USA: Society for Industrial and Applied Mathematics, 2003. P. 255–256.
14. Desmos Graphing Calculator. URL: <https://www.desmos.com/calculator> (accessed: 09.03.2021).
15. Sokolinsky L.B. BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems. *Journal of Parallel and Distributed Computing*. 2021. Vol. 149. P. 193–206. DOI: 10.1016/j.jpdc.2020.12.009.
16. Sahni S., Vairaktarakis G. The master-slave paradigm in parallel computer and industrial settings. *Journal of Global Optimization*. 1996. Vol. 9, no. 3–4. P. 357–377. DOI: 10.1007/BF00121679.
17. Sokolinsky L.B. BSF-skeleton. User manual: arXiv:2008.12256 [cs.DC]. 2020. 22 p.
18. Sokolinsky L.B. BSF-skeleton. 2019. URL: <https://github.com/leonid-sokolinsky/BSF-skeleton> (accessed: 09.03.2021).
19. Gropp W. MPI 3 and Beyond: Why MPI Is Successful and What Challenges It Faces. *Recent Advances in the Message Passing Interface*. EuroMPI 2012. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012. Vol. 7490. P. 1–9. DOI: 10.1007/978-3-642-33518-1_1.
20. Bird R.S. Lectures on Constructive Functional Programming. *Constructive Methods in Computing Science*. NATO ASI Series F: Computer and Systems Sciences. Berlin, Heidelberg: Springer, 1988. Vol. 55. P. 151–216.
21. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources. Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016). CEUR Workshop Proceedings. Vol. 1576. 2016. P. 561–573.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ АЛГОРИТМА TREECODE ДЛЯ РЕШЕНИЯ ЗАДАЧИ N -ТЕЛ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ*

© 2021 А.В. Титов, А.В. Хоперсков

*Волгоградский государственный университет
(400062 Волгоград, пр. Университетский, д. 100)
E-mail: alexandr.titov@volsu.ru, khoperskov@volsu.ru*

Поступила в редакцию: 12.02.2021

Иерархические методы вычисления гравитационных сил для систем N -тел позволяют существенно увеличить качество численного моделирования при решении различных астрофизических задач за счет увеличения числа элементов N , поскольку вместо вычислительной сложности $\sim O(N^2)$ для прямого метода, мы имеем $N \log(N)$ при использовании приближенного метода TreeCode, что позволяет существенно увеличить число частиц в численных моделях. Разработано новое программное обеспечение для решения динамической задачи с большим числом частиц для моделирования галактических бесстолкновительных компонент, в частности, звездной подсистемы и темной массы. В работе представлены результаты тестирования алгоритма TreeCode для параллельной реализации на графических ускорителях NVidia Tesla. Для построения иерархической системы сеток нами реализован быстрый алгоритм построения октодеревьев, основанный на пространственной кривой Мортон. Для оценок качества построенной численной модели используем для сравнения результаты моделирования на основе прямого вычисления сил взаимодействия между всеми N частицами системы. Проведен анализ быстродействия различных реализаций алгоритмов решения задачи N -тел и выполнения интегральных законов сохранения физических характеристик для гравитирующих систем. В частности, проанализированы законы сохранения энергии и момента импульса для вращающегося самогравитирующего диска. Рассмотрены модели с различными критериями оценки удаленности частицы и значениями угла раскрытия θ .

Ключевые слова: задача N -тел, метод Treecode, параллельные вычисления, графические ускорители.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Титов А.В., Хоперсков А.В. Особенности реализации алгоритма Treecode для решения задачи N -тел с использованием графических ускорителей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 53–65. DOI: 10.14529/cmse210204.

Введение

Задача N -тел восходит к небесной механике, где решаются уравнения движения с учетом гравитационного взаимодействия для системы небесных тел [1, 2]. С развитием вычислительной техники стало возможным численное интегрирование уравнений движения первоначально для сотен частиц, что позволило моделировать коллективные процессы в приложении к галактическим системам [3], а с переходом к $N \sim 10^6$ объектом исследований стала метагалактика и первоначально в таких космологических моделях под частицей понималась галактика (см., например, [4]). Современные астрофизические модели N -тел могут содержать свыше 1 млрд элементов (частиц различных видов, моделирующих звезды, темное вещество, газ) [5, 6], и даже превышают 10^{10} элементов в проекте IllustrisTNG. Такие проекты, как Illustris, IllustrisTNG, EAGLE (Evolution and Assembly of GaLaxies and their

*Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2021»

Environments) [5], AGORA [7], FIRE (Feedback In Realistic Environments), Indra, Millennium и Millennium-II Simulations [8] и др. являются своего рода галактическими и космологическими «фабриками» для симуляций эволюции Вселенной на различных масштабах. Результаты таких проектов, например, TNG50, TNG100, TNG300, AGORA выступают в качестве источников исходных данных, которые позволяют на основе обработки таких симуляций изучать механизмы формирования галактических структур, их морфологию и кинематику [9–11].

Модели N -тел активно используются для моделирования широкого круга задач в физике плазмы, физической электронике при описании ионных пучков, динамике разреженных газов [2, 12]. К этому классу моделей относятся методы молекулярной динамики для описания эволюции молекулярных систем с использованием как классических, так и квантовохимических моделей [13–15].

Развитие быстрых методов, в частности, алгоритма Treecode было первоначально связано с моделированием гравитирующих систем [16]. Гравитационное взаимодействие N -тел является ресурсоемкой задачей из-за дальнедействующего характера взаимодействия и использование прямого метода, имеющего вычислительную сложность $O(N^2)$, не позволяет получить решение за приемлемое время для систем с большим количеством частиц. Алгоритм Treecode позволяет снизить вычислительную сложность до $O(N \log N)$ за счет кластеризации удаленных частиц и использования аппроксимации их влияния на каждую частицу [16, 17].

Важным направлением является развитие технологий программирования гибридных вычислительных систем, которая обеспечивает автоматическую адаптацию кода при изменении вычислительной архитектуры [18]. Применение комплексного подхода к решению задач моделирования сложных физических систем с эффективным использованием архитектуры суперкомпьютера является насущной задачей [19]. Массовый переход на вычислительные системы с графическими процессорами NVIDIA Tesla в последние годы для проведения суперкомпьютерных вычислений ставит задачу реализации эффективных кодов для вычисления парных взаимодействий для системы N -тел [15, 20, 21].

В работе описана реализация параллельного кода для графических ускорителей для численного решения динамической задачи для N гравитационно взаимодействующих частиц, что представляет интерес для решения задач, прежде всего, физики галактик. В основе реализованного Treecode-алгоритма лежит построение дерева разрядов посредством преобразования координат частиц к целочисленным индексам с использованием кривой Мортон, описанное в разделе 1. В разделе 2 проведен анализ производительности программной реализации и рассчитаны изменения со временем углового момента и полной энергии системы N -тел. Показана сходимость решений при уменьшении угла раскрытия θ . В заключении приводится краткая сводка результатов, полученных в работе.

1. Модель N -тел и параллельная реализация кода

Рассмотрим систему из N гравитационно взаимодействующих частиц. Уравнения движения для частицы с индексом i ($i = 1, \dots, N$) имеют вид:

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \quad (1)$$

$$\frac{d\mathbf{v}_i}{dt} = -G \sum_{j=1, j \neq i}^N m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j + \varepsilon|^3}, \quad (2)$$

где вектора \mathbf{r} и \mathbf{v} определяют положение и скорость частицы соответственно, m — масса частицы, $\varepsilon \geq 0$ — радиус гравитационного обрезания. Случай $\varepsilon > 0$ позволяет рассматривать бесстолкновительные системы, когда число реальных объектов N_* существенно превышает число частиц N в численной модели. Например, число звезд в типичной крупной галактике $N_* \sim 10^{11}$, а в численной модели мы ограничены, как правило, $N \sim 10^7$. При моделировании систем электрически заряженных частиц указанное различие может быть сколь угодно большим. Для обеспечения бесстолкновительности системы, в которых сила взаимодействия между ближайшими частицами мала по сравнению с вкладом остальных частиц, в численной модели необходимо задавать конечное значение $\varepsilon > 0$ в зависимости от N и пространственной конфигурации изучаемой системы [22, 23].

Ключевая идея алгоритма Treecode заключается в иерархическом разбиении пространства на древовидную структуру. Такая структура позволяет более точно вычислять силы от близкорасположенных частиц, а вклад от более далеких частиц учитывается приближенно так, чтобы иметь контролируемую погрешность суммарной силы. При вычислении силы для каждой частицы происходит рекурсивный обход ячеек дерева и на каждом уровне оценивается удаленность ячейки от частицы с помощью критерия принятия решения. Существуют различные критерии, позволяющие проверить, достаточно ли удалена частица для оценки силы. В работе используются два различных критерия принятия решения: стандартный критерий, введенный в работе [16]:

$$d \geq \frac{l}{\theta}, \quad (3)$$

и модифицированный, который дает более точную оценку силы в случаях, когда центр масс ячейки сильно смещен от ее геометрического центра [24]:

$$d \geq \frac{l}{\theta} + \delta, \quad (4)$$

где d — расстояние от частицы до центра масс ячейки, l — размер ячейки, θ — угол раскрытия (opening angle), δ — расстояние от центра масс ячейки до ее геометрического центра. Параметр θ определяет значение максимального угла, под которым мы видим ячейку из точки, для которой вычисляется гравитационная сила.

На рис. 1 представлена диаграмма потоков разработанного приложения. Параллельный алгоритм включает в себя следующие этапы:

- (1) На первом этапе происходит построение дерева разрядов. Листовые узлы такого дерева содержат упорядоченные по возрастанию индексы частиц, а внутренние узлы разделяют, содержащиеся в нем индексы на два линейных диапазона в соответствии с их общими префиксами в бинарном представлении индексов. Перед построением дерева происходит определение ограничивающего объема посредством вычисления максимального расстояния R_{\max} от центра системы до дальней частицы с использованием алгоритма параллельной редукции. Для всех частиц системы производится преобразование координат частиц к целочисленным значениям, которые используются для вычисления индекса частицы. Индексы частиц определяются на основе пространственной кривой Мортон. Используя двоичное представление кодов Мортон строим

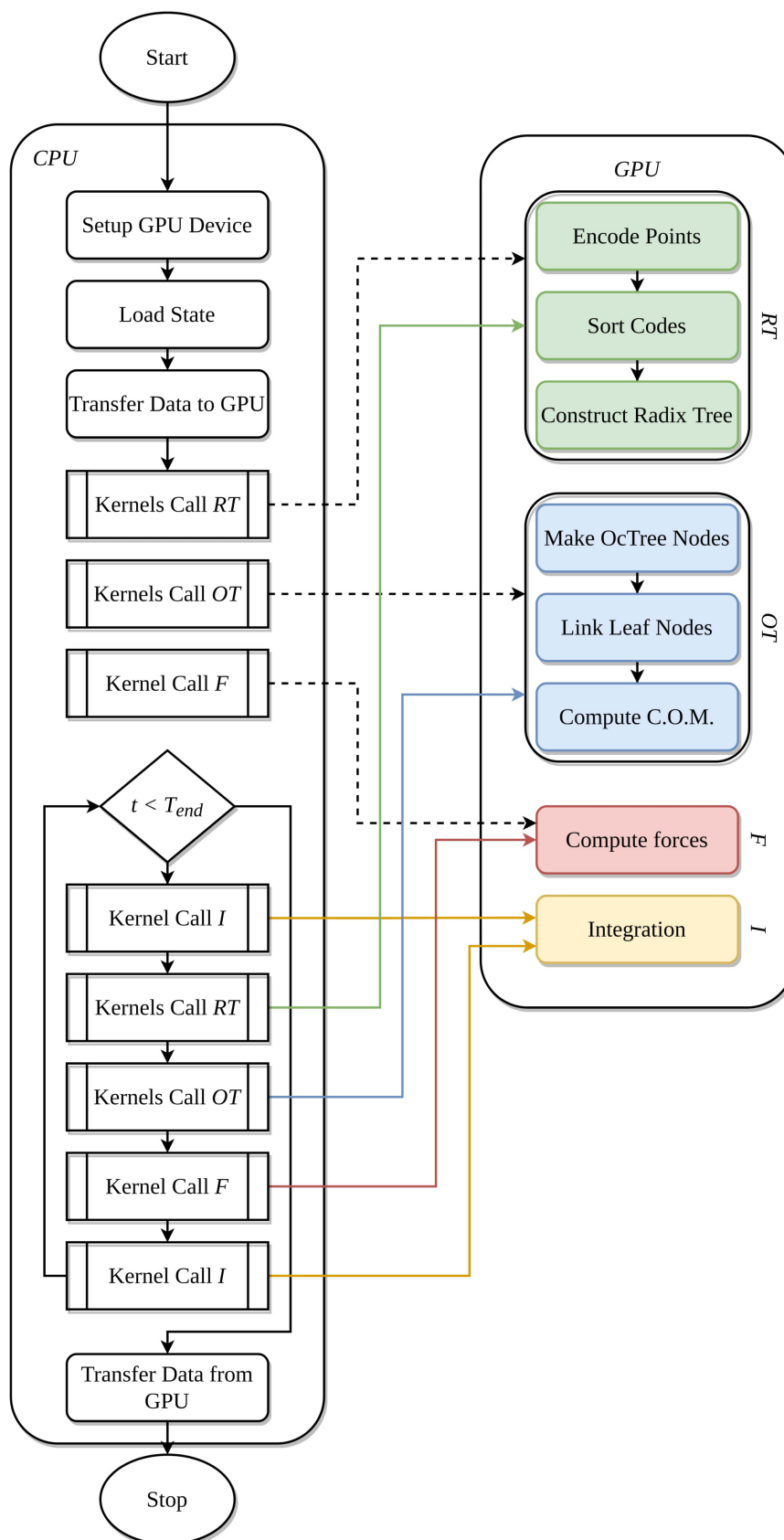


Рис. 1. Диаграмма потоков параллельного приложения

дерево разрядов, являющееся префиксным деревом, в котором внутренние узлы хранят наибольший общий префикс между ключами в соответствующем поддереве [25].

- (2) На втором этапе происходит построение октодеревя. Используя информацию о длинах префиксов в узлах дерева разрядов определяется количество элементов в октодереве [25]. Далее производится поиск родительских узлов октодеревя с использованием ссылки на предков каждого узла дерева разрядов. При привязке частицы к узлу октодеревя эта частица передает ячейке свои характеристики: массу и координаты. Проходя дерево от его листовых узлов к внутренним производим суммирование массы и вычисление центров масс узлов, выполняя синхронизацию потоков после перехода на новый уровень глубины октодеревя.
- (3) Ядро для вычисления сил использует нерекурсивный алгоритм обхода дерева. При проходе узлов октодеревя с использованием одного из критериев принятия решения (3) или (4) определяются необходимость использования аппроксимации.
- (4) Ядро, используемое для интегрирования системы уравнений (1), (2) включает в себя предиктор и корректор. На этапе предиктор рассчитывается скорость и положение частицы по формулам:

$$v_{n+1/2} = v_n + \frac{dv_n}{dt} \frac{dt}{2}, \quad (5)$$

$$x_{n+1} = x_n + v_{n+1/2} dt, \quad (6)$$

где dt — шаг по времени. Затем выполняется перестроение дерева и вычисляются силы с использованием новых положений частиц. На этапе корректор выполняется уточнение скорости в момент времени $n + 1$ по формуле:

$$v_{n+1} = v_{n+1/2} + \frac{dv_{n+1/2}}{dt} \frac{dt}{2}. \quad (7)$$

Все тестовые расчеты проводились для бесстолкновительного гравитирующего трехмерного вращающегося диска. Такой диск является равновесным, находясь на балансе гравитационных, центробежной и сил «давления» (из-за хаотического движения частиц) в радиальном направлении и на балансе гравитационных и сил «давления» в вертикальном направлении (см. подробное описание в [22, 26]). Хаотическое движение частиц в таких бесстолкновительных системах можно характеризовать тремя пространственными распределениями дисперсии скоростей ($c_r(r), c_\varphi(r), c_z(r)$), соответственно, в радиальном, азимутальном и вертикальном направлениях. Мы использовали радиальные профили параметров бесстолкновительного диска из работы [27], пересчитанные в отсутствие темной массы. За счет варьирования начального профиля дисперсии скоростей частиц $c_r(r)$ можно строить как гравитационно устойчивые системы, сохраняющие в целом осесимметричное распределение параметров, так и диски, допускающие развитие гравитационной неустойчивости [22]. В последнем случае в процессе эволюции формируются спиральные структуры значительной амплитуды, что является хорошим тестом для анализа выполнения физических законов сохранения.

2. Анализ производительности и законов сохранения

Параллельные расчеты проводились с использованием графического сопроцессора Nvidia Tesla V100, а тестирование последовательной реализация Treecode [29] проводилось на Intel Xeon E5-2650 v3. Для оценки эффективности распараллеливания проводились сравнения времени вычисления для различных реализаций решения задачи N -тел: параллель-

ная реализация прямого метода (каждая частица с каждой) с использованием программного обеспечения для CUDA [28], открытая версия алгоритма Treecode от Joshua E. Barnes [29].

Были проведены серии расчетов с различным числом частиц и для двух критериев принятия решения. На рис. 2 показаны графики сравнения времени, необходимой для одной итерации вычисления силы для различного числа частиц и различных реализаций решения задачи N -тел: $t_{tc GPU}$ — время вычислений с использованием разработанного параллельного алгоритма для GPUs, $t_d GPU$ — время вычислений с использованием прямого алгоритма на GPUs [28], $t_{tc CPU}$ — время вычислений для последовательной версии Treecode.

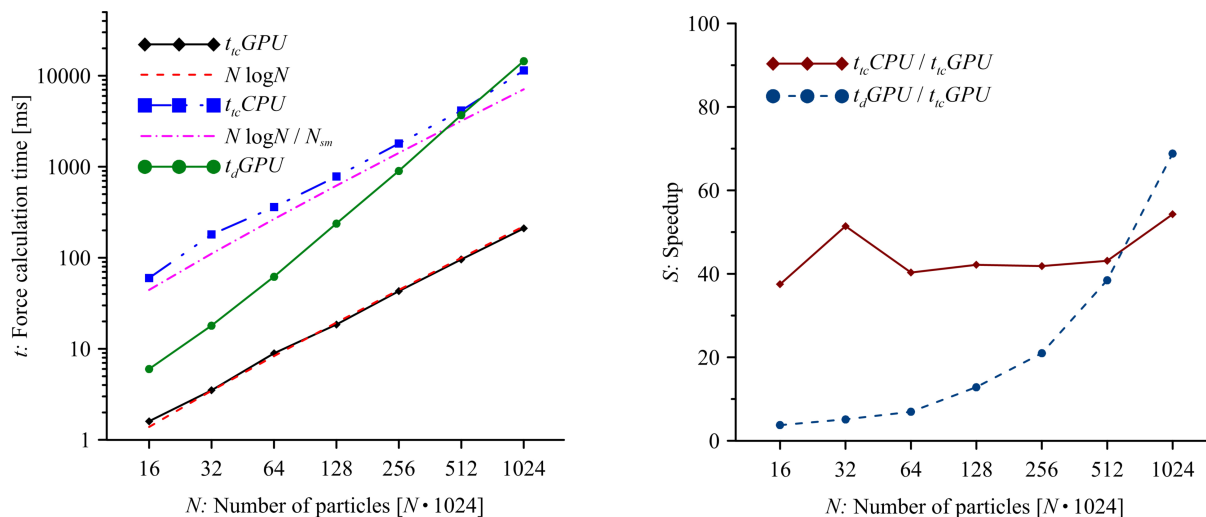


Рис. 2. Сравнение времени расчета силы для различных реализаций с теоретическими оценками (слева) и полученное ускорение S (справа)

На рис. 2 справа представлены результаты ускорения вычислений для модели дисковой галактики [27]. Параллельная реализация Treecode показала ускорение вычислений в 40–45 раз по сравнению с последовательной версией, а также ускорение до 68 раз по сравнению с параллельной реализацией прямого алгоритма при использовании NVidia Tesla V100. Мы провели сравнение точности выполнения основных законов сохранения для выбранной модели диска: углового момента и полной энергии. На рис. 3 и 4 представлены графики изменения относительного отклонения углового момента (слева) и энергии (справа) для различных значений параметра θ .

Относительная погрешность вычисления углового момента варьируется в диапазоне от 0.0001 до 0.008 в зависимости от θ при использовании стандартного критерия, в то время как для модифицированного критерия погрешность варьируется в диапазоне от 0.00002 до 0.002. Относительная погрешность вычисления полной энергии для стандартного критерия лежит в диапазоне от 0.0002 до 0.01, а для модифицированного критерия от 0.0001 до 0.003.

Заключение

Методы молекулярной динамики широко используются для моделирования разнообразных физико-технических объектов и опираются на вычисление парных взаимодействий между частицами [1, 2, 12, 30, 31]. Отметим широкий перечень областей применимости моделей типа N -тел, включая физическую кинетику, физику плазмы и ионных пучков, кос-

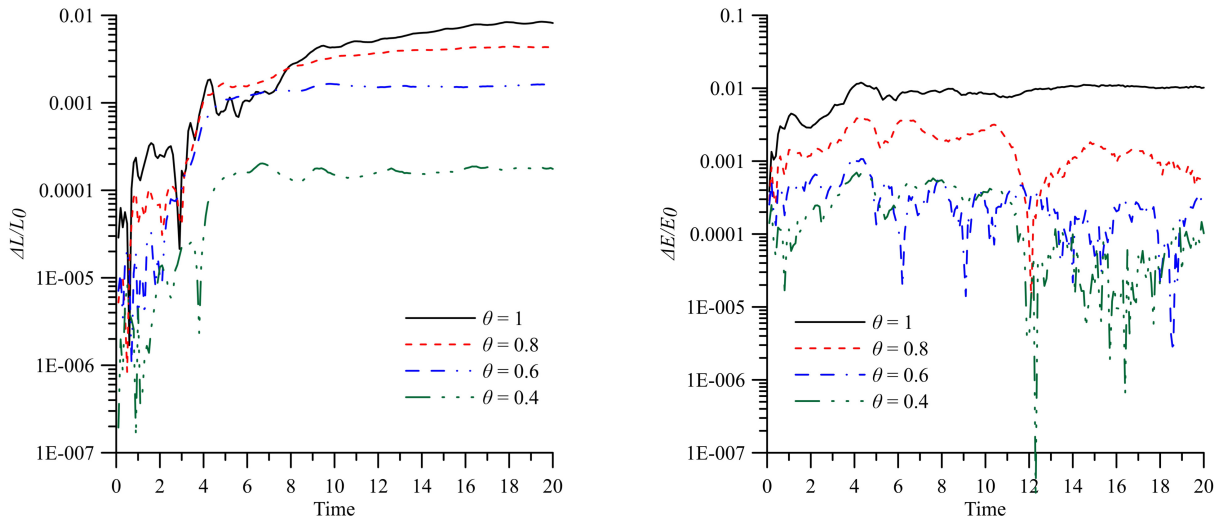


Рис. 3. Изменения относительного отклонения углового момента и энергии для вращающегося диска из $N = 2^{16}$ частиц и стандартного критерия принятия решения

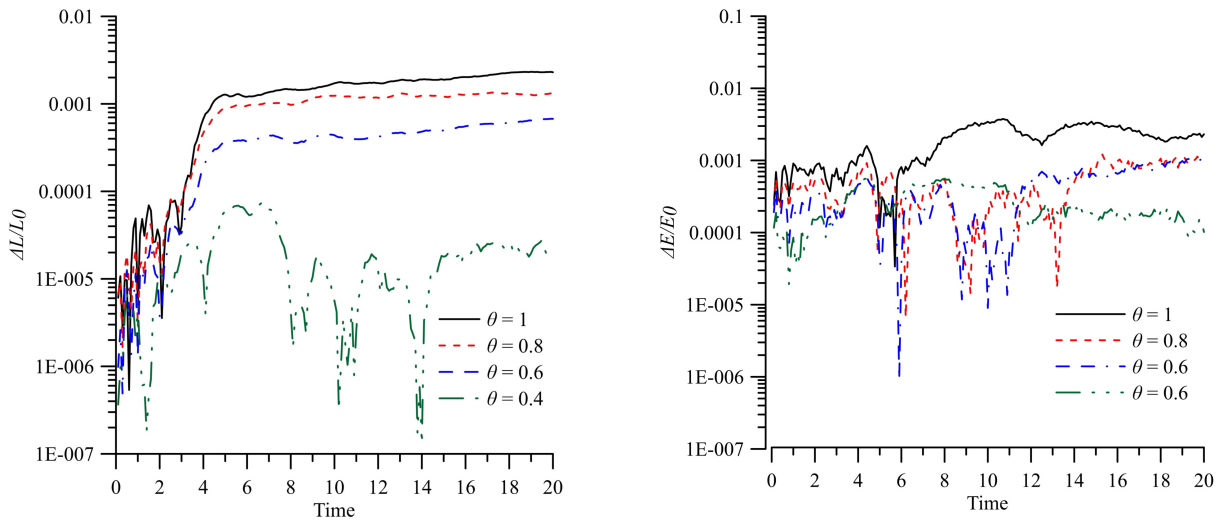


Рис. 4. Изменения относительного отклонения углового момента и энергии для диска с $N = 2^{16}$ при использовании δ критерия

мологию, физику галактик, позволяя моделировать как барионное вещество, так и темную материю в приложении к астрофизическим проблемам [4, 32, 33].

Наш модуль для вычисления суммарной гравитационной силы от большого числа частиц на основе TreeCode можно использовать для моделирования самых различных галактических компонент, включая динамику гравитирующего газа как на больших масштабах ($\sim 10 - 10^4$ пк) [5, 6, 8, 22], так и на малых масштабах ($\sim 0.001 - 10$ пк) [34].

В работе представлены предварительные результаты тестирования разработанного параллельного приложения для моделирования динамики N гравитационно взаимодействующих частиц с использованием графических ускорителей Nvidia Tesla. В качестве модели для тестирования использовался вращающийся бесстолкновительный самогравитирующийся трехмерный диск, первоначальное состояние которого допускало развитие гравитационной неустойчивости, на нелинейной стадии которой формировались центральная перемигачка (так называемая бар-мода) и спиральный узор. Разработанная параллельная реализация алгоритма Treecode для GPUs обеспечивает приемлемое выполнение интегральных законов

сохранения энергии и углового момента в самогравитирующем диске на уровне, сопоставимом с расчетами для CPU.

Производительность программного обеспечения замерялась на графическом сопроцессоре NVidia Tesla V100 (Volta). В условиях проведения тестов величина ускорения составила 45 раз по сравнению с реализацией [29] и до 68 раз по сравнению с параллельным алгоритмом на основе прямого вычисления гравитационной силы [28].

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации №0633-2020-0003. Расчеты проводились на оборудовании ЦКП «Суперкомпьютерный центр коллективного пользования ВолГУ».

Литература

1. Aarseth S.J. The Cambridge N -body lectures. Lecture Notes in Physics. 2008. Vol. 760. 402 p. DOI: 10.1007/978-1-4020-8431-7.
2. Greenspan D. N -body problems and models. World Scientific, 2004. 193 p.
3. Ostriker J.P., Peebles P.J.E. A Numerical Study of the Stability of Flattened Galaxies: or, can Cold Galaxies Survive? // Astrophysical Journal. 1973. Vol. 186. P. 467–480. DOI: 10.1086/152513.
4. Klypin A., Holtzman J., Primack J., Regos E. Structure Formation with Cold plus Hot Dark Matter // Astrophysical Journal. 1993. Vol. 416. P. 1–16. DOI: 10.1086/173210.
5. Schaye J., Crain R.A., Bower R.G. et al. The EAGLE project: simulating the evolution and assembly of galaxies and their environments // Monthly Notices of the Royal Astronomical Society. 2014. Vol. 446, no. 1. P. 521–554. DOI: 10.1093/mnras/stu2058.
6. Pillepich A., Springel V., Nelson D. et al. Simulating galaxy formation with the IllustrisTNG model // Monthly Notices of the Royal Astronomical Society. 2018. Vol. 473, no. 3. P. 4077–4106. DOI: 10.1093/mnras/stx2656.
7. Kim J., Abel T., Agertz O. et al. The AGORA High-resolution Galaxy Simulations Comparison Project // The Astrophysical Journal Supplement. 2014. Vol. 210, id. 14. 20 p. DOI: 10.1088/0067-0049/210/1/14.
8. Boylan-Kolchin M., Springel V., White S.D.M., Jenkins A., Lemson G. Resolving cosmic structure formation with the Millennium-II Simulation // Monthly Notices of the Royal Astronomical Society. 2009. Vol. 398. P. 1150–1164. DOI: 10.1111/j.1365-2966.2009.15191.x.
9. Zhao D., Du M., Ho, L.C., Debattista V.P., Shi J. Barred Galaxies in the IllustrisTNG Simulation // The Astrophysical Journal. 2020. Vol. 904, id. 170. 12 p. DOI: 10.1111/10.3847/1538-4357/abbe1b.
10. Khoperskov S., Zinchenko I., Avramov B. et al. Extreme kinematic misalignment in IllustrisTNG galaxies: the origin, structure, and internal dynamics of galaxies with a large-scale counterrotation // Monthly Notices of the Royal Astronomical Society. 2021. Vol. 500, no. 3. P. 3870–3888. DOI: 10.1093/mnras/staa3330.
11. Ubler H., Genel S., Sternberg A., Genzel R. et al. The kinematics and dark matter fractions of TNG50 galaxies at $z = 2$ from an observational perspective // Monthly Notices of the Royal Astronomical Society. 2021. Vol. 500, no. 4. P. 4597–4619. DOI: 10.1093/mnras/staa3464.

12. Hockney R.W., Eastwood J.W. Computer Simulation Using Particles. CRC Press, 1988, 540 p.
13. Nemukhin A.V., Grigorenko B.L., Savitsky A.P. Computer Modeling of the Structure and Spectra of Fluorescent Proteins // *Acta Naturae*. 2009. Vol. 1, no. 2. P. 33–43. DOI: 10.32607/20758251-2009-1-2-33-43.
14. Холмуродов Х.Т., Алтайский М.В., Пузынин И.В., Дардин Т., Филатов Ф.П. Методы молекулярной динамики для моделирования физических и биологических процессов // *Физика элементарных частиц и атомного ядра*. 2003. Т. 34, № 2. С. 474–515.
15. Суплатов Д.А., Шарапова Я.А., Попова Н.Н., Копылов К.Е., Воеводин Вл.В., Швядас В.К. Молекулярная динамика в силовом поле FF14SB в воде TIP4P-Ew, и в силовом поле FF15IPQ в воде SPC/Eb: сравнительный анализ на GPU и CPU // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2019. Т. 8, № 1. С. 71–88. DOI: 10.14529/cmse190105.
16. Barnes J., Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm // *Nature*. 1986. Vol. 324. P. 446–449. DOI: 10.1038/324446a0.
17. Hamada T., Nitadori K., Benkrid K. et al. A novel multiple-walk parallel algorithm for the Barnes–Hut treecode on GPUs – towards cost effective, high performance N-body simulation // *Computer Science – Research and Development volume*. 2009. Vol. 24. P. 21–31. DOI: 10.1007/s00450-009-0089-1.
18. Gulenok A.A., Dordopulo A.I., Levin I.I., Gudkov V.A. Hybrid computer system programming technology with adaptation and scaling of calculations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. Vol. 6, no. 1. P. 73–86. DOI: 10.14529/cmse170105.
19. Glinskiy B., Kulikov I., Chernykh I., Snytnikov A., Sapetina A., Weins D. The integrated approach to solving large-size physical problems on supercomputers // *Communications in Computer and Information Science*. 2017. Vol. 793. P. 278–289. DOI: 10.1007/978-3-319-71255-0_22.
20. Kuksheva E.A., Malyshkin V.E., Nikitin S.A., Snytnikov A.V., Snytnikov V.N., Vshivkov V.A. Supercomputer simulation of self-gravitating media // *Future Generation Computer Systems*. 2005. Vol. 21, no. 5. P. 749–757. DOI: 10.1016/j.future.2004.05.019.
21. Glinskiy B.M., Kulikov I.M., Snytnikov A.V., Romanenko A.A., Chernykh I.G., Vshivkov V.A. Co-design of parallel numerical methods for plasma physics and astrophysics // *Supercomputing Frontiers and Innovations*. 2014. Vol. 1, no. 3. P. 88–98. DOI: 10.14529/jsfi140305.
22. Fridman A.M., Khoperskov A.V. *Physics of Galactic Disks*. Cambridge International Science Publishing Ltd, 2013. 754 p.
23. Smirnov A.A., Sotnikova N.Ya., Koshkin A.A. Simulations of slow bars in anisotropic disk systems // *Astronomy Letters*. 2017. Vol. 43. P. 61–74. DOI: 10.1134/S1063773717020062.
24. Lukat G., Banerjee R. A GPU accelerated Barnes–Hut tree code for FLASH4 // *New Astronomy*. 2016. Vol. 45. P. 14–28. DOI: 10.1016/j.newast.2015.10.007.

25. Karras T. Maximizing Parallelism in the Construction of BVHs, Octrees, and K-d Trees // Proceedings of the 4th Symposium on High-Performance Graphics. 2012. P. 33–37. DOI: 10.2312/EGGH/HPG12/033-037.
26. Khoperskov A.V., Bizyaev D., Tiurina N., Butenko M. Numerical modelling of the vertical structure and dark halo parameters in disc galaxies // Astronomische Nachrichten. 2010. Vol. 331, no. 7. P. 731–745. DOI: 10.1002/asna.200911402.
27. Khoperskov A.V., Tyurina N.V. A Dynamical Model of the Galaxy // Astronomy Reports. 2003. Vol. 47. P. 443–457. DOI: 10.1134/1.1583771.
28. Khrapov S.S., Khoperskov S.A., Khoperskov A.V. New features of parallel implementation of N -body problems on GPU // Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software. 2018. Vol. 11, no. 1. P. 124–136. DOI: 10.14529/mmp180111.
29. Treecode guide. URL: <https://www.ifa.hawaii.edu/~barnes/treecode/treecode.html> (дата обращения: 30.11.2020).
30. Aarseth S.J. Gravitational N -Body Simulations Tools and Algorithms. Cambridge University Press, 2003. 431 p.
31. Kireyev V., Aichelin J., Bratkovskaya E., Le Fevre A., Lenivenko V., Kolesnikov V., Leifels Y., Voronyuk V. PHQMD Model for the Formation of Nuclear Clusters and Hypernuclei in Heavy Ion // Bulletin of the Russian Academy of Sciences: Physics. 2020. Vol. 84. P. 957–961. DOI: 10.3103/S1062873820080171.
32. Zasov A.V., Moiseev A.V., Khoperskov A.V., Sidorova E.A. Early-type disk galaxies: structure and kinematics // Astronomy Reports. 2008. Vol. 52. P. 79–93. DOI: 10.1134/S1063772908020017.
33. Zasov A.V., Saburova A.S., Khoperskov A.V., Khoperskov S.A. Dark matter in galaxies // Physics-Uspekhi. 2017. Vol. 60, no. 1. P. 3–39. DOI: 10.3367/UFNe.2016.03.037751.
34. Korolev V.V., Eremin M.A., Kovalenko I.G., Zankovich A.M. Numerical simulations of instability in the shell of a supernova remnant expanding in a weakly inhomogeneous interstellar medium // Mathematical Physics and Computer Simulation. 2020. Vol. 23, no. 3. P. 23–35. DOI: 10.15688/mpcm.jvolsu.2020.3.3.

Титов Александр Викторович, аспирант, кафедра информационных систем и компьютерного моделирования, Волгоградский государственный университет (Волгоград, Российская Федерация)

Хоперсков Александр Валентинович, д.ф.-м.н., профессор, кафедра информационных систем и компьютерного моделирования, Волгоградский государственный университет (Волгоград, Российская Федерация)

IMPLEMENTATION FEATURES OF THE TREECODE ALGORITHM FOR SOLVING N -BODY PROBLEMS ON GPUS

© 2021 A.V. Titov, A.V. Khoperskov

Volograd State University (pr. Universitetsky 100, Volograd, 400062 Russia)

E-mail: alexandr.titov@volsu.ru, khoperskov@volsu.ru

Received: 12.02.2021

Hierarchical methods for calculating gravitational forces in a N -body system significantly increase the quality of numerical simulations when solving various astrophysical problems by increasing the number of N elements, since we have the computational complexity $N \log(N)$ for the TreeCode approximate method instead of $\sim O(N^2)$ for the direct method, which allows to greatly increase the number of particles in the models. We developed new software for solving a dynamic problem with a large number of particles for modeling the collisionless components of the galaxies, in particular, stellar subsystem and dark matter. The paper presents the test results for the parallel implementation of the Treecode algorithm for the NVidia Tesla GPUs. To construct a hierarchical grid structure, we implemented a fast parallel octree-construction algorithm based on Morton's space-filling curve. To assess the quality of the constructed numerical model, we use the simulation results based on the direct calculation of the interaction forces between all N particles of the system. We have compared the performance of the different implementations of algorithms for solving the N -body problem and an analysis of the fulfillment of the integral physical conservation laws of a self-gravitational system. The analysis of the fulfillment of the conservation laws of total energy and angular momentum is carried out for a rotating self-gravitating disk. Models with different criteria for a particle remoteness and value of the opening angle θ are considered.

Keywords: N-body problem, Treecode, parallel computing, GPUs.

FOR CITATION

Titov A.V., Khoperskov A.V. Implementation Features of the TreeCode Algorithm for Solving N -body Problems on GPUs. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 53–65. (in Russian) DOI: 10.14529/cmse210204.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Aarseth S.J. The Cambridge N -body lectures. Lecture Notes in Physics. 2008. Vol. 760. 402 p. DOI: 10.1007/978-1-4020-8431-7.
2. Greenspan D. N -body problems and models. World Scientific, 2004. 193 p.
3. Ostriker J.P., Peebles P.J.E. A Numerical Study of the Stability of Flattened Galaxies: or, can Cold Galaxies Survive? *Astrophysical Journal*. 1973. Vol. 186. P. 467–480. DOI: 10.1086/152513.
4. Klypin A., Holtzman J., Primack J., Regos E. Structure Formation with Cold plus Hot Dark Matter. *Astrophysical Journal*. 1993. Vol. 416. P. 1–16. DOI: 10.1086/173210.
5. Schaye J., Crain R.A., Bower R.G. et al. The EAGLE project: simulating the evolution and assembly of galaxies and their environments. *Monthly Notices of the Royal Astronomical Society*. 2014. Vol. 446, no. 1. P. 521–554. DOI: 10.1093/mnras/stu2058.

6. Pillepich A., Springel V., Nelson D. et al. Simulating galaxy formation with the IllustrisTNG model. *Monthly Notices of the Royal Astronomical Society*. 2018. Vol. 473, no. 3. P. 4077–4106. DOI: 10.1093/mnras/stx2656.
7. Kim J., Abel T., Agertz O. et al. The AGORA High-resolution Galaxy Simulations Comparison Project. *The Astrophysical Journal Supplement*. 2014. Vol. 210, id. 14. 20 p. DOI: 10.1088/0067-0049/210/1/14.
8. Boylan-Kolchin M., Springel V., White S.D.M., Jenkins A., Lemson G. Resolving cosmic structure formation with the Millennium-II Simulation. *Monthly Notices of the Royal Astronomical Society*. 2009. Vol. 398. P. 1150–1164. DOI: 10.1111/j.1365-2966.2009.15191.x.
9. Zhao D., Du M., Ho L.C., Debattista V.P., Shi J. Barred Galaxies in the IllustrisTNG Simulation. *The Astrophysical Journal*. 2020. Vol. 904, id. 170. 12 p. DOI: 10.1111/10.3847/1538-4357/abbe1b.
10. Khoperskov S., Zinchenko I., Avramov B. et al. Extreme kinematic misalignment in IllustrisTNG galaxies: the origin, structure, and internal dynamics of galaxies with a large-scale counterrotation. *Monthly Notices of the Royal Astronomical Society*. 2021. Vol. 500, no. 3. P. 3870–3888. DOI: 10.1093/mnras/staa3330.
11. Ubler H., Genel S., Sternberg A., Genzel R. et al. The kinematics and dark matter fractions of TNG50 galaxies at $z = 2$ from an observational perspective. *Monthly Notices of the Royal Astronomical Society*. 2021. Vol. 500, no. 4. P. 4597–4619. DOI: 10.1093/mnras/staa3464.
12. Hockney R.W., Eastwood J.W. *Computer simulation using particles*. Adam Hilger, Bristol, NY: IOP Publishing Ltd, 1988. 540 p.
13. Nemukhin A.V., Grigorenko B.L., Savitsky A.P. *Computer Modeling of the Structure and Spectra of Fluorescent Proteins*. *Acta Naturae*. 2009. Vol. 1, no. 2. P. 33–43. DOI: 10.32607/20758251-2009-1-2-33-43.
14. Kholmurodov K.T., Altaisky M.V., Puzynin I.V., Darden T., Filatov F.P. *Molecular Dynamics Methods for Simulation of Physical and Biological Processes. Physics of Elementary Particles and Atomic Nuclei*. 2003. Vol. 34, no. 2. P. 474–515. (in Russian)
15. Suplatov D.A., Sharapova Y.A., Popova N.N., Kopylov K.E., Voevodin V.I., Svedas V.K. *Molecular Dynamics in the Force Field FF14SB in Water TIP4P-Ew, and in the Force Field FF15IPQ in Water SPC/Eb: a Comparative Analysis on GPU and CPU*. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. Vol. 8, no. 1. P. 71–88. (in Russian) DOI: 10.14529/cmse190105.
16. Barnes J., Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*. 1986. Vol. 324. P. 446–449. DOI: 10.1038/324446a0.
17. Hamada T., Nitadori K., Benkrid K. et al. A novel multiple-walk parallel algorithm for the Barnes–Hut treecode on GPUs – towards cost effective, high performance N -body simulation. *Computer Science – Research and Development volume*. 2009. Vol. 24. P. 21–31. DOI: 10.1007/s00450-009-0089-1.
18. Gulenok A.A., Dordopulo A.I., Levin I.I., Gudkov V.A. Hybrid computer system programming technology with adaptation and scaling of calculations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. Vol. 6, no. 1. P. 73–86. DOI: 10.14529/cmse170105.
19. Glinskiy B., Kulikov I., Chernykh I., Snytnikov A., Sapetina A., Weins D. The integrated approach to solving large-size physical problems on supercomputers. *Communications in Computer and Information Science*. 2017. Vol. 793. P. 278–289. DOI: 10.1007/978-3-319-71255-0_22.

20. Kuksheva E.A., Malyshkin V.E., Nikitin S.A., Snytnikov A.V., Snytnikov V.N., Vshivkov V.A. Supercomputer simulation of self-gravitating media. *Future Generation Computer Systems*. 2005. Vol. 21, no. 5. P. 749–757. DOI: 10.1016/j.future.2004.05.019.
21. Glinskiy B.M., Kulikov I.M., Snytnikov A.V., Romanenko A.A., Chernykh I.G., Vshivkov V.A. Co-design of parallel numerical methods for plasma physics and astrophysics. *Supercomputing Frontiers and Innovations*. 2014. Vol. 1, no. 3. P. 88–98. DOI: 10.14529/jsfi140305.
22. Fridman A.M., Khoperskov A.V. *Physics of Galactic Disks*. Cambridge International Science Publishing Ltd, 2013. 754 p.
23. Smirnov A.A., Sotnikova N.Ya., Koshkin A.A. Simulations of slow bars in anisotropic disk systems. *Astronomy Letters*. 2017. Vol. 43 P. 61–74. DOI: 10.1134/S1063773717020062.
24. Lukat G., Banerjee R. A GPU accelerated Barnes–Hut tree code for FLASH4. *New Astronomy*. 2016. Vol. 45. P. 14–28. DOI: 10.1016/j.newast.2015.10.007.
25. Karras T. Maximizing Parallelism in the Construction of BVHs, Octrees, and K-d Trees. *Proceedings of the 4th Symposium on High-Performance Graphics*. 2012. P. 33–37. DOI: 10.2312/EGGH/HPG12/033-037.
26. Khoperskov A.V., Bizyaev D., Tiurina N., Butenko M. Numerical modelling of the vertical structure and dark halo parameters in disc galaxies. *Astronomische Nachrichten*. 2010. Vol. 331, no. 7. P. 731–745. DOI: 10.1002/asna.200911402.
27. Khoperskov A.V., Tyurina N.V. A Dynamical Model of the Galaxy. *Astronomy Reports*. 2003. Vol. 47. P. 443–457. DOI: 10.1134/1.1583771.
28. Khrapov S.S., Khoperskov S.A., Khoperskov A.V. New features of parallel implementation of N-body problems on GPU. *Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software*. 2018. Vol. 11, no. 1. P. 124–136. DOI: 10.14529/mmp180111.
29. Treecode guide. URL: <https://www.ifa.hawaii.edu/barnes/treecode/treecode.html> (accessed: 30.11.2020).
30. Aarseth S.J. *Gravitational N-Body Simulations Tools and Algorithms*. Cambridge University Press, 2003. 431 p.
31. Kireyeu V., Aichelin J., Bratkovskaya E., Le Fevre A., Lenivenko V., Kolesnikov V., Leifels Y., Voronyuk V. PHQMD Model for the Formation of Nuclear Clusters and Hypernuclei in Heavy Ion. *Bulletin of the Russian Academy of Sciences: Physics*. 2020. Vol. 84. P. 957–961. DOI: 10.3103/S1062873820080171.
32. Zasov A.V., Moiseev A.V., Khoperskov A.V., Sidorova E.A. Early-type disk galaxies: structure and kinematics. *Astronomy Reports*. 2008. Vol. 52. P. 79–93. DOI: 10.1134/S1063772908020017.
33. Zasov A.V., Saburova A.S., Khoperskov A.V., Khoperskov S.A. Dark matter in galaxies. *Advances in Physical Sciences*. 2017. Vol. 60, no. 1. P. 3–39. DOI: 10.3367/UFNe.2016.03.037751.
34. Korolev V.V., Eremin M.A., Kovalenko I.G., Zankovich A.M. Numerical simulations of instability in the shell of a supernova remnant expanding in a weakly inhomogeneous interstellar medium. *Mathematical Physics and Computer Simulation*. 2020. Vol. 23, no. 3. P. 23–35. DOI: 10.15688/mpcm.jvolsu.2020.3.3.

РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ИССЛЕДОВАНИЯ РЕСУРСА ПАРАЛЛЕЛИЗМА ЧИСЛЕННЫХ АЛГОРИТМОВ С ПОМОЩЬЮ ПРОГРАММНОЙ Q-СИСТЕМЫ

© 2021 В.Н. Алеева, П.С. Зотова, Д.С. Склезнев

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: aleevavn@susu.ru, zotovaps2000@gmail.com, skleznevdm@gmail.com

Поступила в редакцию: 16.03.2021

Программная Q -система предназначена для исследования ресурса параллелизма численных алгоритмов. С помощью Q -системы можно вычислить ресурс параллелизма любого численного алгоритма, а также найти алгоритм с лучшим ресурсом параллелизма из множества алгоритмов, решающих алгоритмическую проблему. Теоретической основой Q -системы является концепция Q -детерминанта, где Q — множество операций, используемых алгоритмом. Любой численный алгоритм имеет Q -детерминант и может быть представлен в форме Q -детерминанта. Такое представление является универсальным описанием численных алгоритмов. Q -детерминант состоит из Q -термов. Их число равно числу выходных данных алгоритма. Каждый Q -терм описывает все возможные способы вычисления одного из выходных данных на основе входных данных. Q -детерминант делает алгоритм понятным с точки зрения структуры и реализации. Q -система может быть использована для повышения эффективности параллельных вычислений. В статье описаны новые возможности Q -системы. Новая функциональность позволяет использовать классификацию алгоритмов при их исследовании. Также новая функциональность дает возможность вычислять функции для аппроксимации высоты и ширины алгоритмов более точно и создавать их графическое представление. В результате интерфейс пользователя Q -системы стал удобнее, появилось больше возможностей для исследования ресурса параллелизма алгоритмов.

Ключевые слова: Q-детерминант алгоритма, представление алгоритма в форме Q-детерминанта, Q-эффективная реализация алгоритма, высота алгоритма, ширина алгоритма, ресурс параллелизма алгоритма, программная Q-система.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Алеева В.Н., Зотова П.С., Склезнев Д.С. Расширение возможностей исследования ресурса параллелизма численных алгоритмов с помощью программной Q -системы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 66–81. DOI: 10.14529/cmse210205.

Введение

Одна из самых актуальных проблем — повышение эффективности параллельных вычислений. Эта проблема тесно связана с проблемой использования ресурса параллелизма алгоритмов. Если иметь возможность выявлять ресурс параллелизма алгоритмов, а затем использовать его полностью при разработке программ, то алгоритмы будут реализовываться эффективно. Более того, если иметь возможность сравнивать ресурсы параллелизма алгоритмов, решающих одну и ту же алгоритмическую проблему, а затем использовать алгоритм с лучшим ресурсом параллелизма, то алгоритмическая проблема будет реализована эффективнее. Программная Q -система, созданная на основе концепции Q -детерминанта, предназначена для вычисления и сравнения ресурсов параллелизма численных алгоритмов, поэтому исследование, связанное с расширением ее возможностей, является актуальным.

Целью исследования является расширение возможностей программной Q -системы для исследования ресурса параллелизма численных алгоритмов. Для достижения цели решаются следующие задачи:

- 1) разработка функциональности Q -системы для использования классификации алгоритмов при их исследовании;
- 2) повышение точности аппроксимации функций высоты и ширины алгоритмов;
- 3) реализация графического представления аппроксимации функций высоты и ширины алгоритмов.

Статья организована следующим образом. Раздел 1 содержит обзор работ по теме исследования. В разделе 2 приводятся основные понятия концепции Q -детерминанта. В разделе 3 рассмотрены теоретические основы и программная реализация Q -системы. В разделе 4 приведено описание новой функциональности Q -системы. Заключение содержит краткое изложение полученных результатов, выводы об их применении и описание дальнейшего направления исследований. Статья продолжает исследования, представленные в [1, 7–10].

1. Обзор работ по теме исследования

Наиболее близким к теме исследования, рассматриваемого в статье, является направление, созданное В.В. Воеводиным. Основополагающими работами этого направления являются [2, 15]. В рамках данного направления проводятся очень важные и развитые исследования параллельной структуры алгоритмов и программ для их реализации на параллельных вычислительных системах (ПВС). Эти исследования применяются для создания открытой энциклопедии свойств алгоритмов AlgoWiki [4, 11]. В AlgoWiki для определения ресурса параллелизма алгоритмов используется индивидуальный подход к каждому алгоритму. Единое программное обеспечение для исследования и сравнения ресурсов параллелизма алгоритмов данное направление исследований в настоящее время не содержит.

Исследований, использующих универсальные подходы к решению проблем, связанных с параллельными вычислениями, очень мало. В основном они посвящены разработке параллельных программ. Приведем примеры таких исследований. Т-система [13] является средой разработки с возможностью автоматического динамического распараллеливания программ. Синтез параллельных программ заключается в разработке новых параллельных алгоритмов с использованием базы знаний параллельных алгоритмов для решения более сложных задач. Технология фрагментированного программирования, язык ее реализации и система программирования LuNA разработаны на основе метода синтеза параллельных программ [5]. Чтобы снять ограничения на ресурсы ПВС, рассматривается проектирование параллельных программ с помощью функциональных языков программирования. Работа [12] является примером такого исследования. Приведенные примеры решают проблему построения параллельных программ, но при этом не исследуется ресурс параллелизма алгоритмов, реализуемых программами, и насколько программы его используют.

подавляющее большинство исследований, связанных с параллельными вычислениями, состоят в разработке параллельных программ, реализующих конкретные алгоритмы или учитывающих особенности архитектуры ПВС. Примерами таких исследований являются [14, 16]. Как правило, разработчики параллельных программ не подбирают для решаемой ими задачи алгоритм с лучшим ресурсом параллелизма, а берут некоторый подходящий алгоритм, выбирают некоторую его реализацию, которая их устраивает, не оценивая, является ли эта реализация эффективной.

Данный обзор основан на доступных авторам источниках и, возможно, не является полным. Однако факт, что повышение эффективности параллельных вычислений является одной из самых актуальных проблем, дает основание заключить, что либо инструменты для автоматизированного исследования ресурса параллелизма алгоритмов разработаны, но не достаточно широко известны, либо они не существуют.

2. Концепция Q -детерминанта

Алгоритмическую проблему опишем в виде $\bar{y} = F(N, B)$, где $N = \{n_1, \dots, n_k\}$ — множество параметров размерности или N — пустое множество, B — множество входных данных, $\bar{y} = \{y_1, \dots, y_m\}$ — множество выходных данных, целое число m является либо константой, либо значением вычисляемой функции параметров N , если $N \neq \emptyset$. Значение n_i ($i \in \{1, \dots, k\}$) равно любому положительному целому числу. Если $N = \{n_1, \dots, n_k\}$, то через $\bar{N} = \{\bar{n}_1, \dots, \bar{n}_k\}$ обозначим набор из k положительных целых чисел, где \bar{n}_i — некоторое заданное значение параметра размерности n_i для каждого $i \in \{1, \dots, k\}$. Через $\{\bar{N}\}$ обозначим множество всех возможных k -наборов \bar{N} . Пусть α — алгоритм для решения алгоритмической проблемы, Q — набор операций, используемых алгоритмом α .

Определение 1. Выражение над B и Q определим, как терм в стандартном смысле математической логики [3]. Каждое выражение w имеет уровень вложенности, обозначим его через T^w .

Пример 1. Выражения и их уровни вложенности:

- 1) $w_1 = b_1 \times (b_2 + b_3)/b_4$, $T^{w_1} = 3$;
- 2) $w_2 = (b_1 \geq b_2) \vee ((b_3 \times b_4) \leq b_5 \times (b_6 + b_7))$, $T^{w_2} = 4$.

Определение 2. Выражение называется цепочкой длины n , если оно является результатом применения некоторой ассоциативной операции из Q к n выражениям.

Определение 3. Если $N = \emptyset$, то любое выражение w над B и Q мы называем безусловным Q -термом. Пусть $N \neq \emptyset$ и V — множество всех выражений над B и Q . Тогда любое отображение $w : \{\bar{N}\} \rightarrow V \cup \emptyset$ также называется безусловным Q -термом.

Пусть $N = \emptyset$ и w — безусловный Q -терм. Предположим, что выражение w над B и Q имеет значение логического типа при любой интерпретации переменных B . Тогда безусловный Q -терм w называется безусловным логическим Q -термом. Пусть $N \neq \emptyset$ и w — безусловный Q -терм. Если выражение $w(\bar{N})$ для каждого $\bar{N} \in \{\bar{N}\}$ имеет значение логического типа при любой интерпретации переменных B , то безусловный Q -терм w называется безусловным логическим Q -термом.

Пусть u_1, \dots, u_l — безусловные логические Q -термы, w_1, \dots, w_l — безусловные Q -термы. Тогда множество l пар $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ называется условным Q -термом длины l .

Пусть $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i=1,2,\dots}$ — счетное множество пар безусловных Q -термов. Предположим, что $\{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ — условный Q -терм для любого $l < \infty$. Тогда (\hat{u}, \hat{w}) называется условным бесконечным Q -термом.

Опишем нахождение значения безусловного Q -терма w при интерпретации переменных B . Если $N = \emptyset$, то нахождение значения выражения w означает нахождение значения безусловного Q -терма w при любой интерпретации переменных B . Если $N \neq \emptyset$ и $w(\bar{N}) \neq \emptyset$, то $w(\bar{N})$ является выражением над B и Q . Можно найти значение выражения $w(\bar{N})$ при

любой интерпретации переменных B . Конечно, мы опускаем значение $w(\bar{N}) = \emptyset$. Теперь опишем нахождение значения условного Q -терма (\hat{u}, \hat{w}) при интерпретации переменных B . Пусть $N = \emptyset$. Находим значения выражений u_i, w_i для $i \in \{1, \dots, l\}$. При этом мы можем найти пару u_{i_0}, w_{i_0} такую, что u_{i_0} имеет значение **true**. Следовательно, мы можем найти значение w_{i_0} . Тогда считаем, что (\hat{u}, \hat{w}) имеет значение w_{i_0} . В противном случае считаем, что значение (\hat{u}, \hat{w}) при интерпретации переменных B не определено. Пусть $N \neq \emptyset$ и $\bar{N} \in \{\bar{N}\}$. Находим выражения $u_i(\bar{N}), w_i(\bar{N})$ для $i \in \{1, \dots, l\}$. При этом мы можем найти пару $u_{i_0}(\bar{N}), w_{i_0}(\bar{N})$ такую, что $u_{i_0}(\bar{N})$ имеет значение **true**. Следовательно, мы можем найти значение $w_{i_0}(\bar{N})$. Тогда мы считаем, что (\hat{u}, \hat{w}) имеет значение $w_{i_0}(\bar{N})$. В противном случае считаем, что значение (\hat{u}, \hat{w}) для \bar{N} и при такой интерпретации переменных B не определено. Аналогично можно определить значение условного бесконечного Q -терма.

Определение 4. Пусть $M = \{1, \dots, m\}$ и алгоритм α состоит в нахождении для каждого $i \in M$ значения y_i путем вычисления значения Q -терма f_i . Тогда набор Q -термов $\{f_i \mid i \in M\}$ называется Q -детерминантом алгоритма α . Система уравнений $\{y_i = f_i \mid i \in M\}$ называется представлением алгоритма α в форме Q -детерминанта.

Определение 5. Процесс вычисления Q -термов $\{f_i \mid i \in M\}$ алгоритма α называется реализацией алгоритма α . Если реализация алгоритма α состоит в том, что некоторые операции выполняются одновременно, то она называется параллельной реализацией.

Для дальнейшего изложения будем использовать некоторое разбиение множества M . Предположим, что U, C и I образуют разбиение множества $M = \{1, \dots, m\}$ с пустыми членами, то есть: $U \cup C \cup I = M$; $U \cap C = U \cap I = C \cap I = \emptyset$; кроме того, одно или два подмножества U, C и I могут быть пустыми. Предположим также, что подмножества U, C и I можно связать с подмножествами множества Q -термов $\{f_i \mid i \in M\}$ таким образом:

- 1) для каждого $i \in U$ существует Q -терм f_i , который является безусловным, и $f_i = w^i$;
- 2) для каждого $i \in C$ существует Q -терм f_i , который является условным, и $f_i = \{(u_j^i, w_j^i)\}_{j \in \{1, \dots, l(i)\}}$, где $l(i)$ — либо константа, либо значение вычисляемой функции от N , если $N \neq \emptyset$;
- 3) для каждого $i \in I$ существует Q -терм f_i , который является условным бесконечным, и $f_i = \{(u_j^i, w_j^i)\}_{j \in \{1, 2, \dots\}}$.

Определение 6. Опишем реализацию алгоритма α , называемую Q -эффективной.

Пусть $N = \emptyset$. Зададим интерпретацию переменных B . Будем вычислять выражения

$$W = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_j^i, w_j^i(i \in I, j \in \{1, 2, \dots\})\} \quad (1)$$

одновременно, параллельно. Мы говорим, что операция готова к выполнению, если вычислены значения всех ее операндов. При вычислении каждого из выражений W (см. (1)) мы выполняем операции, как только они готовы к выполнению. Если несколько операций цепочки готовы к выполнению, то они выполняются по схеме сдваивания. Например, вычисление цепочки $a_1 + a_2 + a_3 + a_4$ по схеме сдваивания выполняется так: сначала вычисляем $b_1 = a_1 + a_2$ и $b_2 = a_3 + a_4$ одновременно, затем $c = b_1 + b_2$. Если для $i \in C \cup I$ и $j \in \{1, 2, \dots\}$ u_j^i имеет значение **false**, то вычисление соответствующего w_j^i прекращается. Если для $i \in C \cup I$ и $j \in \{1, 2, \dots\}$ вычисление пары (u_j^i, w_j^i) приводит к тому, что значение одного выражения не определено, то вычисление второго выражения прекращается. Если

для $i \in C \cup I$ вычисление некоторой пары $(u_{j_0}^i, w_{j_0}^i)$ приводит к определению их значений и значение $u_{j_0}^i$ **true**, то вычисление выражений u_j^i, w_j^i прекращается для любого $j \neq j_0$.

Теперь пусть $N \neq \emptyset$. Зададим интерпретацию переменных B и $\bar{N} \in \{\bar{N}\}$. Получаем множество выражений

$$W(\bar{N}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in I, j \in \{1, 2, \dots\})\}. \quad (2)$$

Выражения $W(\bar{N})$ (см. (2)) вычисляются по аналогии с выражениями W (см. (1)).

Определение 7. Реализация алгоритма α называется выполнимой, если одновременно должно выполняться конечное (непустое) множество операций.

Замечание 1. Есть алгоритмы, Q -эффективная реализация которых невыполнима.

Пример 2. Алгоритм для вычисления суммы S ряда $\sum_{m=1}^{\infty} (-1)^m \frac{1}{m}$ с заданной точностью ϵ имеет следующее представление в форме Q -детерминанта

$$S = \left\{ \left(\frac{1}{2} < \epsilon, -1 \right), \left(\frac{1}{3} < \epsilon, -1 + \frac{1}{2} \right), \dots, \left(\frac{1}{m} < \epsilon, -1 + \frac{1}{2} - \dots + (-1)^{m-1} \frac{1}{m-1} \right), \dots \right\}.$$

Первый уровень вложенности выражений содержит счетное множество операций деления, поэтому Q -эффективная реализация алгоритма не является выполнимой.

Q -эффективная реализация полностью использует ресурс параллелизма алгоритма, который опишем далее. Для этого будем использовать следующие условия и обозначения.

1. Алгоритм α представлен в форме Q -детерминанта $y_i = f_i$ для всех $i \in M$.
2. Значения Q -термов f_i для всех $i \in M$ определяются при любой интерпретации переменных B и для любого $\bar{N} \in \{\bar{N}\}$, если $N \neq \emptyset$.
3. Пусть $N = \emptyset$ и $I \neq \emptyset$. Тогда при заданной интерпретации переменных B для любого $i \in I$ существует пара выражений $u_{j_i}^i, w_{j_i}^i$ такая, что значение $u_{j_i}^i$ равно **true** и значение $w_{j_i}^i$ определено. Введем обозначение

$$\widetilde{W} = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_{j_i}^i, w_{j_i}^i(i \in I)\}.$$

4. Пусть $N \neq \emptyset$ и $I \neq \emptyset$. Тогда при заданной интерпретации переменных B для любого $\bar{N} \in \{\bar{N}\}$ и $i \in I$ существует пара выражений $u_{j_i}^i(\bar{N}), w_{j_i}^i(\bar{N})$ такая, что значение $u_{j_i}^i(\bar{N})$ равно **true**, а значение $w_{j_i}^i(\bar{N})$ определено. Введем обозначение

$$\widetilde{W}(\bar{N}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_{j_i}^i(\bar{N}), w_{j_i}^i(\bar{N})(i \in I)\}.$$

5. Q -эффективная реализация алгоритма α выполнима.

Введем понятия высоты и ширины алгоритма α , которые характеризуют его ресурс параллелизма.

Определение 8. Обозначим через D_α высоту и через P_α ширину алгоритма α и определим их следующим образом.

Если $N = \emptyset$, то

$$D_\alpha = \begin{cases} \max_{w \in W} T^w & \text{если } I = \emptyset, \\ \max_{w \in \widetilde{W}} T^w & \text{если } I \neq \emptyset; \end{cases} \quad (3)$$

$$P_\alpha = \max_{1 \leq r \leq D_\alpha} \sum_{w \in W} O_r^w, \quad (4)$$

где O_r^w — количество операций уровня вложенности r выражения w .

Если $N \neq \emptyset$, то

$$D_\alpha(\bar{N}) = \begin{cases} \max_{w(\bar{N}) \in W(\bar{N})} T^{w(\bar{N})} & \text{если } I = \emptyset, \\ \max_{w(\bar{N}) \in \widetilde{W}(\bar{N})} T^{w(\bar{N})} & \text{если } I \neq \emptyset; \end{cases} \quad (5)$$

$$P_\alpha(\bar{N}) = \max_{1 \leq r \leq D_\alpha(\bar{N})} \sum_{w(\bar{N}) \in W(\bar{N})} O_r^{w(\bar{N})}, \quad (6)$$

где $O_r^{w(\bar{N})}$ — количество операций уровня вложенности r выражения $w(\bar{N})$.

В формулах (3), (4), (5), (6) обозначения W и $W(\bar{N})$ те же, что в (1) и (2) соответственно. Содержательный смысл формул (3) и (4) следующий. Если параметров размерности нет и Q -детерминант не содержит условные бесконечные Q -термы, то высота алгоритма — это максимальный уровень вложенности выражений W , а ширина алгоритма — максимальное количество операций одного уровня вложенности выражений W . В случае, если параметров размерности нет, но Q -детерминант содержит условные бесконечные Q -термы, то высота алгоритма — это максимальный уровень вложенности выражений \widetilde{W} , а ширина алгоритма — максимальное количество операций одного уровня вложенности выражений W , не превосходящего высоту алгоритма. Содержательный смысл формул (5) и (6) можно описать аналогично. D_α характеризует время выполнения Q -эффективной реализации алгоритма, а P_α — количество процессоров, необходимое для выполнения Q -эффективной реализации алгоритма. Понятия D_α и P_α соотносятся с понятиями высоты и ширины минимальной ярусно-параллельной формы алгоритма.

3. Q -система

3.1. Теоретические основы Q -системы

Теоретические основы Q -системы составляют методы исследования ресурса параллелизма численных алгоритмов, разработанные с помощью концепции Q -детерминанта:

- 1) метод построения Q -детерминанта алгоритма;
- 2) метод получения Q -эффективной реализации алгоритма;
- 3) метод вычисления характеристик ресурса параллелизма алгоритма;
- 4) метод сравнения характеристик ресурса параллелизма алгоритмов.

Опишем эти методы и поясним их на простых примерах.

Q -детерминанты алгоритмов можно построить с помощью блок-схем алгоритмов. При этом блок-схемы используют блоки «терминал», «решение», «процесс», «данные» и имеют ограничения, описанные в работе [9]. Для построения Q -детерминанта алгоритма его блок-схема анализируется при фиксированных значениях параметров размерности алгоритмической проблемы N , если $N \neq \emptyset$.

Если Q -детерминант алгоритма имеет безусловные Q -термы, то проход по блок-схеме идет последовательно в соответствии с выполнением алгоритма. Для каждого $i \in U$ из содержимого блоков «процесс», участвующих в вычислении f_i , формируются выражения w^i , если $N = \emptyset$ и $w^i(\bar{N})$, если $N \neq \emptyset$. Если Q -детерминант содержит условные Q -термы, то блок-схема имеет хотя бы один блок «решение» с условием, содержащим входные данные. Прохождение такого блока порождает ветвление. После прохода по каждой из ветвей для всех $i \in C$ и $j \in \{1, \dots, l(i)\}$ получаются пары выражений (u_j^i, w_j^i) , если $N = \emptyset$ и $(u_j^i(\bar{N}), w_j^i(\bar{N}))$, если $N \neq \emptyset$ для всех условных Q -термов f_i . Выражения u_j^i , если $N = \emptyset$ и $u_j^i(\bar{N})$, если $N \neq \emptyset$ формируются из условий блоков «решение», которые содержат входные данные, а w_j^i , если $N = \emptyset$ и $w_j^i(\bar{N})$, если $N \neq \emptyset$ на основе содержимого блоков «процесс». Условные бесконечные Q -термы содержат Q -детерминанты итерационных алгоритмов. Ограничив количество итераций, можно свести случай Q -детерминанта с условными бесконечными Q -термами к случаю Q -детерминанта с условными Q -термами. Для ограничения количества итераций используется параметр L . Его значение обозначим через \bar{L} . Оно может принимать целые положительные значения. \bar{L} задает длину Q -термов f_i для $i \in I$.

Пример 3. Рассмотрим алгоритм вычисления скалярного произведения векторов $\vec{a}^1 = (a_1^1, \dots, a_n^1)$ и $\vec{a}^2 = (a_1^2, \dots, a_n^2)$, не использующий схему сдваивания. В этом случае $N \neq \emptyset$ и $I = \emptyset$. С помощью метода построения Q -детерминанта для любого $n \in \{2, 3, \dots\}$ может быть получено представление алгоритма в форме Q -детерминанта

$$(\vec{a}^1, \vec{a}^2) = (\dots (a_1^1 \times a_1^2 + a_2^1 \times a_2^2) + a_3^1 \times a_3^2) + \dots + a_n^1 \times a_n^2).$$

В разделе 2 определено, что при Q -эффективной реализации алгоритма выполняется одновременное вычисление выражений W (см. (1)), если $N = \emptyset$ и выражений $W(\bar{N})$ (см. (2)), если $N \neq \emptyset$, при этом операции выполняются по мере вычисления их операндов. Если $N = \emptyset$ и $I \neq \emptyset$, то вместо выражений W будем использовать выражения

$$W(\bar{L}) = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_j^i, w_j^i(i \in I, j \in \{1, \dots, \bar{L}\})\}.$$

Если $N \neq \emptyset$ и $I \neq \emptyset$, то вместо выражений $W(\bar{N})$ будем использовать выражения

$$W(\bar{N}, \bar{L}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in I, j \in \{1, \dots, \bar{L}\})\}.$$

Метод получения Q -эффективной реализации алгоритма заключается в вычислении уровней вложенности операций, входящих в выражения: W , если $N = \emptyset$ и $I = \emptyset$; $W(\bar{L})$, если $N = \emptyset$ и $I \neq \emptyset$; $W(\bar{N})$, если $N \neq \emptyset$ и $I = \emptyset$; $W(\bar{N}, \bar{L})$, если $N \neq \emptyset$ и $I \neq \emptyset$.

Пример 4. Для алгоритма из примера 3 используем множество $W(\bar{N})$, состоящее из выражения $(\dots (a_1^1 \times a_1^2 + a_2^1 \times a_2^2) + a_3^1 \times a_3^2) + \dots + a_n^1 \times a_n^2$. Операции умножения имеют уровень вложенности 1, уровень вложенности i -й ($i \in \{1, \dots, n-1\}$) операции сложения равен $i+1$.

Вычисление характеристик ресурса параллелизма алгоритма α выполняется по следующим правилам. Если $N = \emptyset$ и $I = \emptyset$, то высота и ширина алгоритма α не зависят от параметров N и L . Высота обозначается D_α и вычисляется по формуле (3). Ширина обозначается P_α и вычисляется по формуле (4). Если $N = \emptyset$ и $I \neq \emptyset$, то высота и ширина алгоритма α зависят от параметра L . Значение высоты алгоритма обозначается $D_\alpha(\bar{L})$ и вычисляется

по формуле $D_\alpha(\bar{L}) = \max_{w \in W(\bar{L})} T^w$. Значение ширины алгоритма обозначается $P_\alpha(\bar{L})$ и вычисляется по формуле $P_\alpha(\bar{L}) = \max_{1 \leq r \leq D_\alpha(\bar{L})} \sum_{w \in W(\bar{L})} O_r^w$. Если $N \neq \emptyset$ и $I = \emptyset$, то высота и ширина алгоритма α зависят от параметров N . Значение высоты алгоритма обозначается $D_\alpha(\bar{N})$ и вычисляется по формуле (5). Значение ширины алгоритма обозначается $P_\alpha(\bar{N})$ и вычисляется по формуле (6). Если $N \neq \emptyset$ и $I \neq \emptyset$, то высота и ширина алгоритма α зависят от параметров N и L . Значение высоты алгоритма обозначается $D_\alpha(\bar{N}, \bar{L})$ и вычисляется по формуле $D_\alpha(\bar{N}, \bar{L}) = \max_{w(\bar{N}) \in W(\bar{N}, \bar{L})} T^{w(\bar{N})}$. Значение ширины алгоритма обозначается $P_\alpha(\bar{N}, \bar{L})$ и вычисляется по формуле $P_\alpha(\bar{N}, \bar{L}) = \max_{1 \leq r \leq D_\alpha(\bar{N}, \bar{L})} \sum_{w(\bar{N}) \in W(\bar{N}, \bar{L})} O_r^{w(\bar{N})}$.

Пример 5. Высота алгоритма из примера 3 должна вычисляться по формуле (5), а ширина по формуле (6), следовательно, высота и ширина равны n .

Допустим, что алгоритмы α и β решают одну и ту же алгоритмическую проблему. Метод сравнения характеристик ресурса параллелизма алгоритмов сравнивает высоту и ширину алгоритмов α и β , если они удовлетворяют одному из четырех наборов условий.

1. Пусть $N = \emptyset$ и $I = \emptyset$. Тогда D_α , P_α , D_β и P_β определены.
2. Пусть $N = \emptyset$ и $I \neq \emptyset$. Тогда
 - 1) $D_\alpha(\bar{L})$ и $P_\alpha(\bar{L})$ определены, если $\bar{L} \in V_\alpha(L)$;
 - 2) $D_\beta(\bar{L})$ и $P_\beta(\bar{L})$ определены, если $\bar{L} \in V_\beta(L)$;
 - 3) $V_\alpha(L) \cap V_\beta(L) \neq \emptyset$.
3. Пусть $N \neq \emptyset$ и $I = \emptyset$. Тогда
 - 1) $D_\alpha(\bar{N})$ и $P_\alpha(\bar{N})$ определены, если $\bar{N} \in V_\alpha(N)$;
 - 2) $D_\beta(\bar{N})$ и $P_\beta(\bar{N})$ определены, если $\bar{N} \in V_\beta(N)$;
 - 3) $V_\alpha(N) \cap V_\beta(N) \neq \emptyset$.
4. Пусть $N \neq \emptyset$ и $I \neq \emptyset$. Тогда
 - 1) $D_\alpha(\bar{N}, \bar{L})$ и $P_\alpha(\bar{N}, \bar{L})$ определены, если $(\bar{N}, \bar{L}) \in V_\alpha(N, L)$;
 - 2) $D_\beta(\bar{N}, \bar{L})$ и $P_\beta(\bar{N}, \bar{L})$ определены, если $(\bar{N}, \bar{L}) \in V_\beta(N, L)$;
 - 3) $V_\alpha(N, L) \cap V_\beta(N, L) \neq \emptyset$.

Пример 6. Поясним метод сравнения характеристик ресурса параллелизма алгоритмов на примере алгоритма вычисления скалярного произведения векторов, не использующего схему сдваивания (см. пример 3), и алгоритма вычисления скалярного произведения векторов $\vec{a}^1 = (a_1^1, \dots, a_n^1)$ и $\vec{a}^2 = (a_1^2, \dots, a_n^2)$, в котором схема сдваивания применяется. В качестве алгоритма α используем первый алгоритм, а в качестве алгоритма β второй. С помощью метода построения Q -детерминанта для любого значения $n \in \{2, 3, \dots\}$ может быть получено представление алгоритма β в форме Q -детерминанта $(\vec{a}^1, \vec{a}^2) = a_1^1 \times a_1^2 + a_2^1 \times a_2^2 + \dots + a_n^1 \times a_n^2$. Так как алгоритм β использует схему сдваивания, то $D_\beta = 1 + \lceil \log_2 n \rceil$, $P_\beta = n$. Пусть высота и ширина алгоритма α определены для значений размерности множества $V_\alpha(N) = \{2, 3, 4, 5, 9, 15, 20, 50\}$, а высота и ширина алгоритма β для значений размерности множества $V_\beta(N) = \{2, 4, 5, 7, 9, 20, 35, 50\}$. Тогда $V_\alpha(N) \cap V_\beta(N) = \{2, 4, 5, 9, 20, 50\} \neq \emptyset$. Алгоритмы α и β удовлетворяют набору условий 3, поэтому к ним можно применить метод сравнения.

С помощью метода сравнения вычисляются значения ΔD и ΔP по правилам.

1. Если $N = \emptyset$ и $I = \emptyset$, то $\Delta D = D_\alpha - D_\beta$, $\Delta P = P_\alpha - P_\beta$.

2. Если $N = \emptyset$ и $I \neq \emptyset$, то

$$\Delta D = \sum_{\bar{L} \in (V_\alpha(L) \cap V_\beta(L))} (D_\alpha(\bar{L}) - D_\beta(\bar{L})), \quad \Delta P = \sum_{\bar{L} \in (V_\alpha(L) \cap V_\beta(L))} (P_\alpha(\bar{L}) - P_\beta(\bar{L})).$$

3. Если $N \neq \emptyset$ и $I = \emptyset$, то

$$\Delta D = \sum_{\bar{N} \in (V_\alpha(N) \cap V_\beta(N))} (D_\alpha(\bar{N}) - D_\beta(\bar{N})), \quad \Delta P = \sum_{\bar{N} \in (V_\alpha(N) \cap V_\beta(N))} (P_\alpha(\bar{N}) - P_\beta(\bar{N})).$$

4. Если $N \neq \emptyset$ и $I \neq \emptyset$, то

$$\Delta D = \sum_{(\bar{N}, \bar{L}) \in (V_\alpha(N, L) \cap V_\beta(N, L))} (D_\alpha(\bar{N}, \bar{L}) - D_\beta(\bar{N}, \bar{L})),$$

$$\Delta P = \sum_{(\bar{N}, \bar{L}) \in (V_\alpha(N, L) \cap V_\beta(N, L))} (P_\alpha(\bar{N}, \bar{L}) - P_\beta(\bar{N}, \bar{L})).$$

По результатам вычислений делаются следующие выводы. Если $\Delta D < 0$, то высота алгоритма α меньше высоты алгоритма β . Если $\Delta D = 0$, то алгоритмы имеют одинаковую высоту. Если $\Delta D > 0$, то высота алгоритма α больше высоты алгоритма β . Если $\Delta P < 0$, то ширина алгоритма α меньше ширины алгоритма β . Если $\Delta P = 0$, то алгоритмы имеют одинаковую ширину. Если $\Delta P > 0$, то ширина алгоритма α больше ширины алгоритма β .

Пример 7. Применим метод сравнения для алгоритмов α и β из примера 6. Используя правило 3, получим $\Delta D = 63$, $\Delta P = 0$. Следовательно, можно сделать вывод, что высота алгоритма α больше высоты алгоритма β , а ширина алгоритмов одинакова.

3.2. Программная реализация Q-системы

Q-система состоит из двух подсистем: для создания Q-детерминантов и для вычисления ресурса параллелизма численных алгоритмов. Первая подсистема реализует метод построения Q-детерминанта алгоритма на основе его блок-схемы. Во второй подсистеме реализованы остальные методы исследования ресурса параллелизма численных алгоритмов. Диаграмма развертывания Q-системы представлена на рис. 1.

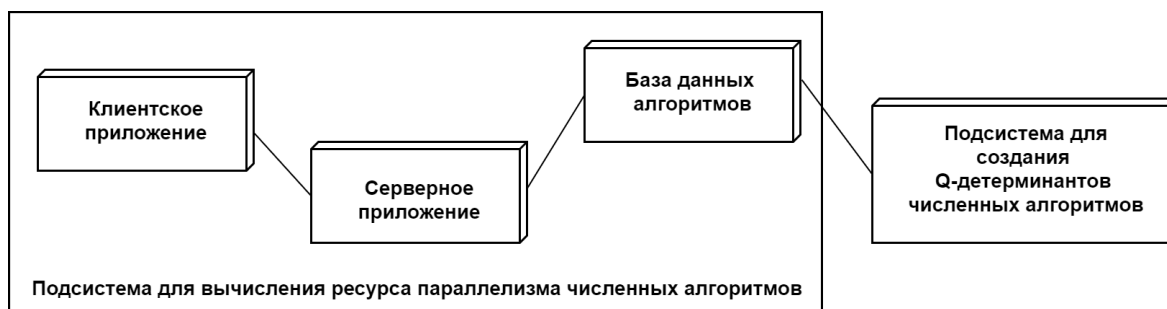


Рис. 1. Диаграмма развертывания Q-системы

Подсистема для создания Q-детерминантов численных алгоритмов является .NET-приложением, разработанным на объектно-ориентированном языке программирования C#. Формат обмена данными JSON применяется для описания входных и выходных данных подсистемы. Формат описания и примеры блок-схем алгоритмов и выходных файлов приведены в работах [7–9].

Подсистема для вычисления ресурса параллелизма численных алгоритмов включает базу данных, серверное и клиентское приложения. Для разработки использовалась СУБД **PostgreSQL**. База данных состоит из таблиц **Algorithms** и **Determinants**. Таблица **Algorithms** имеет атрибуты: идентификатор, имя, описание алгоритма, количество Q -детерминантов алгоритма, загруженных в базу данных. Атрибуты таблицы **Determinants**: идентификатор Q -детерминанта, идентификатор алгоритма, значения параметров размерности N , если $N \neq \emptyset$, иначе 0, Q -детерминант, значение D_α , значение P_α , значение параметра L , если $I \neq \emptyset$, иначе 0. Серверное приложение реализует методы: запись нового алгоритма, обновление информации об алгоритме, получение списка алгоритмов с полной информацией о них, сравнение характеристик ресурса параллелизма алгоритмов, удаление алгоритма и его Q -детерминантов, загрузка Q -детерминанта и характеристик ресурса параллелизма, получение списка Q -детерминантов, скачивание Q -детерминанта, удаление Q -детерминанта. На главной странице клиентского приложения находится список алгоритмов, записанных в базу данных. Также на ней показано количество Q -детерминантов, загруженных в базу данных для каждого алгоритма, расположены элементы интерфейса, которые позволяют любому пользователю сравнить алгоритмы, перейти на страницу со списком Q -детерминантов, загруженных в базу данных для выбранного алгоритма, аппроксимировать функции $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$. Страница со списком Q -детерминантов содержит для каждого Q -детерминанта значения параметров размерности \bar{N} , количество итераций \bar{L} , значения $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$. Интерфейс для авторизованных пользователей обеспечивает добавление, редактирование и удаление алгоритмов, загрузку и удаление Q -детерминантов.

Пример 8. В настоящее время в Q -системе описано десять специально подобранных алгоритмов с различными структурами Q -детерминантов, в том числе алгоритмы α и β (см. пример 6), используемых для тестирования функциональности Q -системы. Для алгоритмов α и β в базу данных загружено соответственно 36 и 78 Q -детерминантов. При загрузке каждого Q -детерминанта выполнялось вычисление и запись в базу данных значений высоты и ширины. Результатом сравнения алгоритмов α и β будет вывод, приведенный в примере 7.

4. Расширение функциональности Q -системы

4.1. Обзор алгоритмов в соответствии с их классификацией

В процессе эксплуатации Q -системы количество алгоритмов в базе данных может расти. В этом случае использование общего списка алгоритмов будет не всегда удобным, поэтому был реализован способ обзора алгоритмов в соответствии с их классификацией. Возможность обзора алгоритмов в виде общего списка сохранена.

Примером использования при исследовании алгоритмов их классификации является открытая энциклопедия свойств алгоритмов AlgoWiki [4]. Под классификацией алгоритмов в нашем исследовании мы понимаем следующий принцип систематизации алгоритмов. Исследуемые алгоритмы делятся на классы, классы могут иметь подклассы первого уровня, которые могут иметь подклассы второго уровня и так далее. Количество классов, подклассов и уровней подклассов не ограничено. Один и тот же алгоритм может принадлежать нескольким классам или подклассам. Критерии распределения алгоритмов по классам и подклассам должны определять их исследователи — пользователи Q -системы.

Программное обеспечение, разработанное для обзора алгоритмов в соответствии с их классификацией, позволяет создать для классификатора систему папок и подпапок, соот-

ветствующих классам и подклассам классификатора. Структуры строк, соответствующих алгоритмам, при обзоре на основе иерархической классификации и с помощью общего списка одинаковы. Таким образом, при обзоре алгоритмов в соответствии с классификацией для работы с алгоритмами используется та же функциональность Q -системы, что и при обзоре алгоритмов в виде общего списка. Например, внутри любого класса или подкласса алгоритмов можно проводить сравнение характеристик ресурса параллелизма двух алгоритмов, переходить на страницу для аппроксимации функций высоты и ширины, на страницу с Q -детерминантами выбранного алгоритма. Авторизованный пользователь имеет право производить операции добавления, редактирования и удаления элементов классификации алгоритмов, записывать новые алгоритмы, редактировать и удалять существующие.

Пример 9. Для алгоритмов, описанных в Q -системе, был создан классификатор и осуществлена его программная поддержка. В результате обзор алгоритмов можно осуществлять с использованием следующей системы папок и подпапок.

1. Папка «**1. Решение СЛАУ**» содержит две подпапки:
 - подпапка «**1.1. Прямые методы решения СЛАУ**» содержит метод Гаусса—Жордана с выбором в качестве ведущего первого ненулевого элемента ведущей строки;
 - подпапка «**1.2. Итерационные методы решения СЛАУ**» содержит методы Гаусса—Зейделя и Якоби.
2. Папка «**2. Умножение матриц**» содержит два алгоритма: не использующий и использующий схему сдваивания.
3. Папка «**3. Скалярное произведение векторов**» содержит два алгоритма: не использующий и использующий схему сдваивания.
4. Папка «**4. Вычисление максимального элемента в массиве**» содержит алгоритм поиска максимального элемента в массиве чисел.
5. Папка «**5. Решение квадратных уравнений**» содержит два алгоритма: без использования операции унарного минуса и с использованием операции унарного минуса.

4.2. Аппроксимация характеристик ресурса параллелизма алгоритмов и ее графическое представление

Характеристики ресурса параллелизма алгоритмов $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$ хранятся в базе данных в табличном виде, то есть их значения известны только для некоторых наборов значений параметров размерности и количества итераций. Вместе с тем при исследовании ресурса параллелизма алгоритмов важно иметь возможность вычислять характеристики ресурса параллелизма при любых значениях \bar{N} и \bar{L} . Поэтому важно решить задачу аппроксимации, а точнее, интерполяции и экстраполяции функций высоты $D(\bar{N}, \bar{L})$ и ширины $P(\bar{N}, \bar{L})$. Для наглядности целесообразно использовать кроме аналитического представления аппроксимирующей функции ее графическое представление.

Мы разработали новую функциональность Q -системы для аппроксимации характеристик ресурса параллелизма алгоритмов $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$ и графического представления аппроксимирующих функций. При описании и использовании данной функциональности параметры размерности и количества итераций обозначаются через $\{x_i\}_{i=1, \dots, k}$. Сначала перечисляются $k - 1$ параметров размерности в том порядке, в каком они хранятся в базе данных, переменная x_k обозначает количество итераций. Если $\bar{L} = 0$, то x_k не используется.

Ранее в составе Q -системы было разработано программное обеспечение для аппроксимации характеристик ресурса параллелизма алгоритмов с помощью полиномиальных

функций, зависящих от $\{x_i\}_{i=1,\dots,k}$. При этом степень аппроксимирующей полиномиальной функции вводилась пользователем по запросу системы. Это программное обеспечение не обеспечивало достаточную точность аппроксимации при больших значениях параметров в случае, если характеристика ресурса параллелизма зависела от $\log_2(x_i)$, где $x_i \in \{x_i\}_{i=1,\dots,k}$. Графическое представление аппроксимирующих функций реализовано не было.

Для реализации новой функциональности Q -системы для аппроксимации характеристик ресурса параллелизма алгоритмов и графического представления аппроксимирующих функций было разработано программное обеспечение, которое для любого алгоритма, исследуемого с помощью Q -системы, решает следующие задачи.

1. Вычисление аппроксимирующих полиномиальных функций для высоты $D(\bar{N}, \bar{L})$ и ширины $P(\bar{N}, \bar{L})$ с помощью метода наименьших квадратов и минимизации невязки. В процессе вычисления подбирается степень и тип полинома. Выбор типа аппроксимирующей полиномиальной функции производится между полиномиальной функцией, зависящей от $\{x_i\}_{i=1,\dots,k}$, и полиномиальной функцией, зависящей от $\{\log_2(x_i)\}_{i=1,\dots,k}$.
2. Создание графического представления аппроксимирующих функций для высоты и ширины при условии, что они зависят не более чем от двух параметров $\{x_i\}_{i=1,\dots,k}$. Графическое представление содержит график аппроксимирующей функции, дополненный изображениями точек из базы данных.
3. Расчет значений аппроксимирующих функций для любых значений параметров $\{x_i\}_{i=1,\dots,k}$, заданных пользователем.

Для аппроксимации функций высоты и ширины алгоритма пользователь должен выбрать алгоритм из списка и кликнуть на элементе со знаком суммы Σ . В результате будет выведена страница с аппроксимацией для выбранного алгоритма, содержащая аппроксимирующие функции и их графические представления, если они были созданы. На этой же странице пользователь может задать значения параметров $\{x_i\}_{i=1,\dots,k}$ и получить вычисленные с помощью аппроксимирующих функций значения высоты и ширины алгоритма.

На рис. 2 и рис. 3 показано графическое представление аппроксимирующих функций высоты и ширины алгоритма β (см. пример 6) и метода Гаусса—Жордана с выбором в качестве ведущего первого ненулевого элемента ведущей строки. Оценка $3x_1$ высоты метода Гаусса—Жордана, где x_1 — размер матрицы СЛАУ, ранее была доказана в [1] математически. Рисунки 2 и 3 получены с помощью Q -системы.

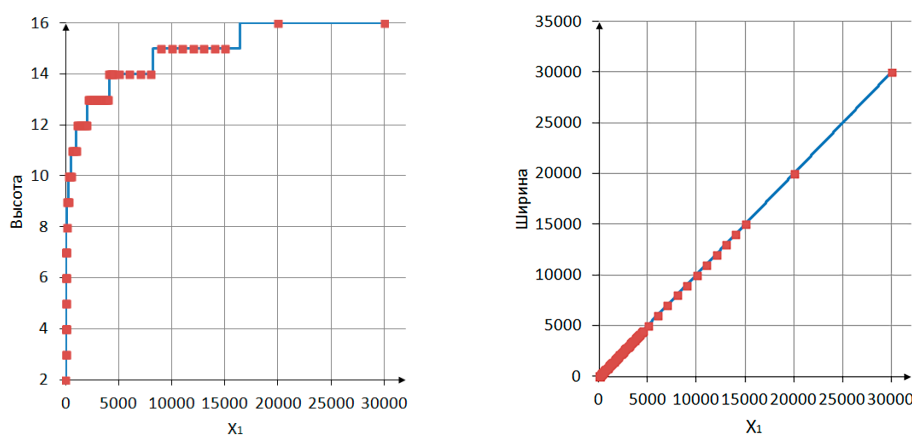


Рис. 2. Графическое представление аппроксимирующих функций высоты и ширины алгоритма вычисления скалярного произведения векторов

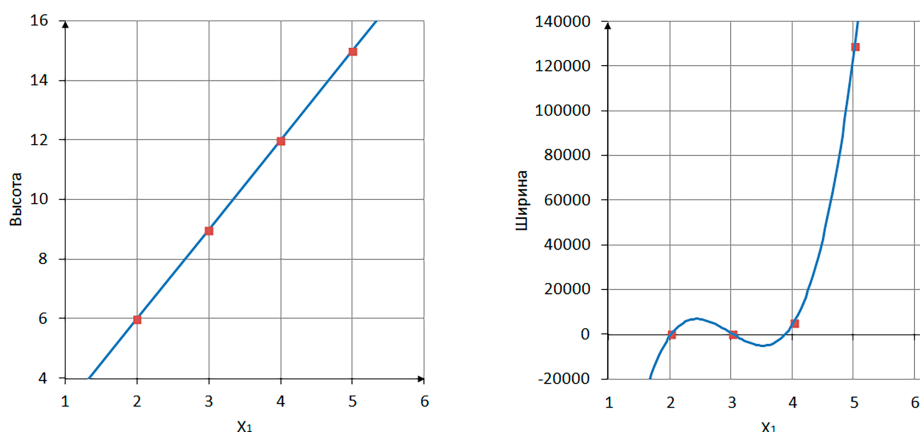


Рис. 3. Графическое представление аппроксимирующих функций высоты и ширины метода Гаусса–Жордана

Заключение

В статье описано расширение возможностей программной Q -системы для исследования ресурса параллелизма численных алгоритмов в результате решения следующих задач:

- 1) разработана функциональность Q -системы для использования классификации алгоритмов при их исследовании;
- 2) повышена точность аппроксимации функций высоты и ширины алгоритмов;
- 3) реализовано графическое представление аппроксимации функций высоты и ширины алгоритмов.

Полученные результаты дают пользователям больше возможностей для исследования ресурса параллелизма алгоритмов, а также делают интерфейс пользователя более удобным. Q -система, метод проектирования Q -эффективных программ [6] и технология Q -эффективного программирования [7] в совокупности решают проблему повышения эффективности параллельных вычислений. Планируется дальнейшее расширение возможностей Q -системы, а также ее доведение до промышленной эксплуатации. Q -система свободно доступна в сети Интернет по адресу <https://qclient.herokuapp.com>, что позволяет ознакомиться с ее возможностями, для которых авторизация не требуется.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00865 а и при поддержке Правительства РФ в соответствии с Постановлением № 211 от 16.03.2013 г. (соглашение № 02.А03.21.0011).

Литература

1. Алеева В.Н. Анализ параллельных численных алгоритмов. Препринт № 590. Новосибирск: ВЦ СО АН СССР, 1985. 23 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
3. Ершов Ю.Л., Палотин Е.А. Математическая логика. М.: Наука, 1987. 336 с.
4. Открытая энциклопедия свойств алгоритмов. URL: <https://algowiki-project.org/ru> (дата обращения: 06.03.2021).
5. Akhmed-Zaki D., Lebedev D., Malyshev V. et al. Automated Construction of High Performance Distributed Programs in LuNA System // Parallel Computing Technologies (PaCT 2019). Lecture Notes in Computer Science. 2019. Vol. 11657. P. 3–9. DOI: 10.1007/978-3-030-25636-4_1.

6. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant // Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
7. Aleeva V.N. Improving Parallel Computing Efficiency // Proceedings – 2020 Global Smart Industry Conference (GloSIC 2020). IEEE, 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
8. Aleeva V.N., Alev R.Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms // Proceedings – 2018 Global Smart Industry Conference (GloSIC 2018). IEEE, 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.
9. Aleeva V., Bogatyreva E., Skleznev A. et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism // Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
10. Aleeva V.N., Sharabura I.S., Suleymanov D.E. Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q -determinant // Parallel Computing Technologies (PaCT 2015). Lecture Notes in Computer Science. 2015. Vol. 9251. P. 3–9. DOI: 10.1007/978-3-319-21909-7_1.
11. Antonov A.S., Dongarra J., Voevodin V.V. AlgoWiki Project as an Extension of the Top500 Methodology // Supercomputing Frontiers and Innovations. 2018. Vol. 5, no. 1. P. 4–10. DOI: 10.14529/jsfi180101.
12. Legalov A.I., Vasilyev V.S., Matkovskii I.V. et al. A Toolkit for the Development of Data-Driven Functional Parallel Programmes // Parallel Computational Technologies (PCT'2018). Communications in Computer and Information Science. 2018. Vol. 910. P. 16–30. DOI: 10.1007/978-3-319-99673-8_2.
13. Moskovsky A., Roganov V., Abramov S. Parallelism Granules Aggregation with the T-System // Parallel Computing Technologies (PaCT 2007). Lecture Notes in Computer Science. 2007. Vol. 4671. P. 293–302. DOI: 10.1007/978-3-540-73940-1_30.
14. Rajashri A. Parallelization of shortest path algorithm using OpenMP and MPI // International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017. P. 304–309. DOI: 10.1109/I-SMAC.2017.8058360.
15. Voevodin V.V., Voevodin V.I. The V-Ray technology of optimizing programs to parallel computers // Numerical Analysis and Its Applications (WNAA 1996). Lecture Notes in Computer Science. 1997. Vol. 1196. P. 546–556. DOI: 10.1007/3-540-62598-4_136.
16. Wang Q., Liu J., Tang X. et al. Accelerating embarrassingly parallel algorithm on Intel MIC // IEEE International Conference on Progress in Informatics and Computing. 2014. P. 213–218. DOI: 10.1109/PIC.2014.6972327.

Алеева Валентина Николаевна, к.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Зотова Полина Сергеевна, студент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Склезнев Дмитрий Сергеевич, студент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

ADVANCEMENT OF RESEARCH FOR THE PARALLELISM RESOURCE OF NUMERICAL ALGORITHMS WITH THE HELP OF SOFTWARE Q -SYSTEM

© 2021 V.N. Aleeva, P.S. Zotova, D.S. Skleznev

South Ural University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: alevavn@susu.ru, zotovaps2000@gmail.com, skleznevdm@gmail.com

Received: 16.03.2021

The software Q -system is designed to study the parallelism resource of numerical algorithms. By use of Q -system, we can calculate the parallelism resource of any numerical algorithm. Also, for the set of algorithms solving a given algorithmic problem, we can find an algorithm with the best parallelism resource. The theoretical basis of the Q -system is the concept of Q -determinant where Q is the set of operations used by the algorithm. Any numerical algorithm has a Q -determinant and can be represented in the form of a Q -determinant. Such a representation is a universal description of numerical algorithms. The Q -determinant consists of Q -terms. Their number is equal to the number of output data items. Each Q -term describes all possible ways to calculate one of the output data items based on the input data. Q -determinant makes the algorithm transparent in terms of structure and implementation. The Q -system can be used for the improvement of parallel computing efficiency. In this paper we describe new features of the Q -system. With the new functionality, we can use the classification of algorithms when studying them. Also with the new functionality, we can calculate the functions for approximating the height and width of the algorithms more precisely and create a graphical representation of them. As a result, we have a more convenient user interface of the Q -system and more opportunities to study the parallelism resource of algorithms.

Keywords: Q -determinant of algorithm, representation of algorithm in form of Q -determinant, Q -effective implementation of algorithm, algorithm height, algorithm width, parallelism resource of algorithm, software Q -system.

FOR CITATION

Aleeva V.N., Zotova P.S., Skleznev D.S. Advancement of Research for the Parallelism Resource of Numerical Algorithms with the Help of Software Q -system. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 66–81. (in Russian) DOI: 10.14529/cmse210205.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Aleeva V.N. Analysis of Parallel Numerical Algorithms. Preprint no. 590. Novosibirsk, Computing Center of the Siberian Branch of the Academy of Sciences of the USSR, 1985. 23 p. (in Russian)
2. Voevodin V.V., Voevodin V.I. Parallel Computing. St.Petersburg, BHV-Petersburg, 2002. 608 p. (in Russian)
3. Ershov Yu.L., Palyutin E.A. Mathematical Logic. Moscow, Mir, 1984. 303 p.
4. Open Encyclopedia of Parallel Algorithmic Features. URL: <https://algowiki-project.org/en> (accessed: 06.03.2021).

5. Akhmed-Zaki D., Lebedev D., Malyshkin V. et al. Automated Construction of High Performance Distributed Programs in LuNA System. Parallel Computing Technologies (PaCT 2019). Lecture Notes in Computer Science. 2019. Vol. 11657. P. 3–9. DOI: 10.1007/978-3-030-25636-4_1.
6. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant. Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
7. Aleeva V.N. Improving Parallel Computing Efficiency. Proceedings – 2020 Global Smart Industry Conference (GloSIC 2020). IEEE, 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
8. Aleeva V.N., Alev R.Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms. Proceedings – 2018 Global Smart Industry Conference (GloSIC 2018). IEEE, 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.
9. Aleeva V., Bogatyreva E., Skleznev A. et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism. Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
10. Aleeva V.N., Sharabura I.S., Suleymanov D.E. Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q -determinant. Parallel Computing Technologies (PaCT 2015). Lecture Notes in Computer Science. 2015. Vol. 9251. P. 3–9. DOI: 10.1007/978-3-319-21909-7_1.
11. Antonov A.S., Dongarra J., Voevodin V.V. AlgoWiki Project as an Extension of the Top500 Methodology. Supercomputing Frontiers and Innovations. 2018. Vol. 5, no. 1. P. 4–10. DOI: 10.14529/jsfi180101.
12. Legalov A.I., Vasilyev V.S., Matkovskii I.V. et al. A Toolkit for the Development of Data-Driven Functional Parallel Programmes. Parallel Computational Technologies (PCT'2018). Communications in Computer and Information Science. 2018. Vol. 910. P. 16–30. DOI: 10.1007/978-3-319-99673-8_2.
13. Moskovsky A., Roganov V., Abramov S. Parallelism Granules Aggregation with the T-System. Parallel Computing Technologies (PaCT 2007). Lecture Notes in Computer Science. 2007. Vol. 4671. P. 293–302. DOI: 10.1007/978-3-540-73940-1_30.
14. Rajashri A. Parallelization of shortest path algorithm using OpenMP and MPI. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017. P. 304–309. DOI: 10.1109/I-SMAC.2017.8058360.
15. Voevodin V.V., Voevodin V.I. The V-Ray technology of optimizing programs to parallel computers. Numerical Analysis and Its Applications (WNAA 1996). Lecture Notes in Computer Science. 1997. Vol. 1196. P. 546–556. DOI: 10.1007/3-540-62598-4_136.
16. Wang Q., Liu J., Tang X. et al. Accelerating embarrassingly parallel algorithm on Intel MIC. IEEE International Conference on Progress in Informatics and Computing. 2014. P. 213–218. DOI: 10.1109/PIC.2014.6972327.

КВАНТОВО-ХИМИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕРМОХИМИЧЕСКИХ СВОЙСТВ МОЛЕКУЛ ВЫСОКОЭНЕРГЕТИЧЕСКИХ ТЕТРАЗИНОВ*

© 2021 В.М. Волохов¹, Е.С. Амосова¹, А.В. Волохов¹, Т.С. Зюбина¹,
Д.Б. Лемперт¹, Л.С. Яновский^{1,2,3}, Д.А. Варламов¹

¹Институт проблем химической физики РАН

(142432 Московская обл., Черноголовка, пр. им. акад.Семенова, д. 1),

²Центральный институт авиационного моторостроения им. П.И. Баранова

(111116 Москва, ул. Авиамоторная, д. 2),

³Московский авиационный институт

(125993 Москва, Волоколамское шоссе, д. 4)

E-mail: {vvt, aes, vav, zhubin, lempert, Yanovskiy, dima}@icp.ac.ru

Поступила в редакцию: 11.05.2021

В статье приведены результаты расчетов структурных и термодинамических характеристик ряда потенциально высокоэнергетических соединений: $C_2N_6O_4$, $C_2N_6O_5$, $C_2N_6O_6$, $C_2H_2N_6O_4$, $C_3HN_7O_6$, $C_3HN_7O_4F_2$, $C_4N_{10}O_{12}$, $C_3HN_6O_4F$, $C_4N_{10}O_8F_4$, $C_4N_8O_8F_2$ с использованием квантово-химических *ab initio* методов. Квантово-химическое моделирование выполнено с применением прикладного программного комплекса GAUSSIAN 09. С использованием гибридного функционала плотности B3LYP проведен поиск оптимальной геометрии молекул, рассчитаны ИК-спектры поглощения, структурные параметры и смещения атомов для наиболее интенсивных колебаний. Методом атомизации получены величины энтальпий образования (ЭО). Расчеты выполнены с использованием комбинированных методов G4(MP2) и G4. Продемонстрирована зависимость ЭО от структуры молекулы. Приведены примеры использованных вычислительных конфигураций на базе нескольких вычислительных ресурсов, описаны некоторые особенности проведенных расчетов. Сделано сопоставление временных затрат и погрешностей при использовании разных методов на одинаковых структурах.

Ключевые слова: квантово-химические расчеты, пакет Gaussian, энтальпия образования, высокоэнергетические вещества, ИК-спектры газообразных молекул, комбинированные методы G4 и G4(MP2).

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Волохов В.М., Амосова Е.С., Волохов А.В., Зюбина Т.С., Лемперт Д.Б., Яновский Л.С., Варламов Д.А. Квантово-химическое моделирование термодинамических свойств молекул высокоэнергетических тетразинов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 82–96. DOI: 10.14529/cmse210206.

Введение

Большой интерес к созданию высокоэнергетических материалов различного назначения стимулируется бурно развивающимися новыми технологиями, особенно в области двигателей перспективных летательных аппаратов. В последнее время в создании новых материалов с определенными свойствами все более важную роль играют современные компьютерные технологии [1–3].

Поскольку одним из основных критериев энергоемкости химического соединения является величина его стандартной энтальпии образования ΔH_f° , то определение этой

* Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2021»

величины, как экспериментальной, так и расчетной, становится ключевой задачей для оценки эффективности применения того или иного вещества как компонента высокоэнергетических материалов. В задачах термохимии важно знать, как меняется энтальпия образования (ЭО) вещества при некоторых изменениях структуры. Одним из наиболее надежных методов исследования описанных выше зависимостей ЭО от различных параметров, характеризующих молекулы, является квантово-химические *ab initio* расчеты с использованием высокоточных составных методов, таких как G4 и G4(MP2), входящих в состав пакета Gaussian 09 (<https://gaussian.com>).

Настоящая работа посвящена определению квантово-химическим методом ΔH_f° ряда высокоэнтальпийных производных с общим строением, показанным на рис. 1.

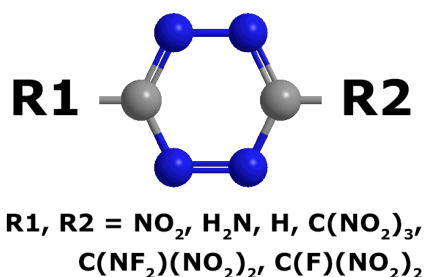


Рис. 1. Структура исследуемых молекул

Статья организована следующим образом. В разделе 1 рассмотрена методика моделирования структурных и термохимических свойств изученных веществ в целом и описаны способы расчета конкретных параметров. В разделе 2 приведены основные результаты моделирования и сделано их обсуждение, в разделе 3 освещены основные аспекты собственно вычислительных процедур и описаны ресурсы, на которых проводились расчеты. В Заключении сформулированы основные выводы как теоретического, так и практического характера, и приводится краткая сводка результатов работы.

1. Методика расчета

Расчет энтальпии образования исследуемых газообразных молекул был проведен методом атомизации аналогично предыдущим работам авторов [4, 5]. Моделирование было выполнено в рамках программного комплекса GAUSSIAN 09 [6]: поиск оптимальной геометрии и расчет ИК-спектров производился с использованием хорошо зарекомендовавшего себя в молекулярных расчетах гибридного функционала плотности B3LYP [7, 8] с базисом 6-311+G(2d,p), а для дальнейшего расчета энтальпии образования были использованы комбинированные методы G4(MP2) и G4 [9, 10]. Метод G4 из серии методов Gaussian-N (Gn), разработанной специально для термохимических расчетов, был представлен в 2007 году Кертиссом и коллегами [9]. Для аппроксимации энергий более точных расчетов в комбинированном методе G4 используются расчеты энергий как метода CCSD(T) с высоким уровнем корреляции электронов и базисным набором среднего размера (6-31G(d)), так и более низкого уровня (например, MP4 и MP2) с большими базисными наборами. В методе G4(MP2), представляющем собой модификацию метода G4, расчет MP4 заменяется на MP2, что делает данный метод существенно менее требовательным к компьютерным ресурсам.

Ниже приведены основные этапы расчета энтальпии образования соединения с общей формулой $C_wH_xN_yO_zF_p$ методом атомизации:

1. Расчет энергии атомизации в нерелятивистском приближении

$$\sum D_0 = wE_0(C) + xE_0(H) + yE_0(N) + zE_0(O) + pE_0(F) - E_0(C_wH_xN_yO_zF_p)$$

где $E_0(C)$, $E_0(H)$, $E_0(N)$, $E_0(O)$, $E_0(F)$ — рассчитанные полные энергии атомов. Полную энергию молекулы $E_0(C_wH_xN_yO_zF_p)$ рассчитывали по формуле $E_0(C_wH_xN_yO_zF_p) = \varepsilon_0 + ZPE$, где ε_0 — полная энергия молекулы, а ZPE — сумма энергий нулевых колебаний всех колебательных мод молекулы.

2. Расчет энтальпии образования при 0 К

$\Delta H_f^0(C_wH_xN_yO_zF_p, 0K) = w\Delta H_f^0(C, 0K) + x\Delta H_f^0(H, 0K) + y\Delta H_f^0(N, 0K) + z\Delta H_f^0(O, 0K) + p\Delta H_f^0(F, 0K) - \sum D_0$
Первые пять слагаемых — это энтальпии образования газообразных атомных компонент, известные из эксперимента.

3. Расчет энтальпии образования при 298.15 К

$$\begin{aligned} \Delta H_f^0(C_wH_xN_yO_zF_p, 298K) = & \Delta H_f^0(C_wH_xN_yO_zF_p, 0K) + (H^0(C_wH_xN_yO_zF_p, 298K) - H^0(C_wH_xN_yO_zF_p, 0K)) - \\ & - w(H^0(C, 298K) - H^0(C, 0K)) - x(H^0(H, 298K) - H^0(H, 0K)) - \\ & - y(H^0(N, 298K) - H^0(N, 0K)) - z(H^0(O, 298K) - H^0(O, 0K)) - \\ & - p(H^0(F, 298K) - H^0(F, 0K)) \end{aligned}$$

Второе слагаемое получается из расчета молекулы. С третьего по седьмой члены в последнем уравнении известны из эксперимента (или рассчитаны по экспериментальным молекулярным постоянным). Значения энтальпии образования газообразных атомов и термические поправки могут быть взяты из различных справочников или литературных источников, например [11–15].

В настоящей работе мы использовали экспериментальные атомные энтальпии образования из термодинамических таблиц NIST-JANAF [12]. Поскольку теоретический расчет систематически завышает величины частот нулевых колебаний, делается коррекция частот с использованием эмпирически подобранных коэффициентов. Для получения более точных частот необходимо корректировать частоты колебаний при расчете поправок ZPE и поправки ($H^0(C_wH_xN_yO_zF_p, 298K) - H^0(C_wH_xN_yO_zF_p, 0K)$). Для этого используются значения масштабирующих множителей, рекомендуемые в литературе для различных методов расчетов и различных базисных наборов [6].

2. Результаты и обсуждение

В табл. 1 приведены структуры рассчитанных молекул, молекулярные веса, коэффициент насыщенности вещества окислителем (альфа), энтальпии образования (в ккал/моль, кДж/моль, кДж/кг), полученные на разных уровнях расчета: G4(MP2) и G4. На рис. 2 приведен график изменения энтальпий образования рассчитанных газообразных молекул. Из рис. 2 и табл. 1 видно, что использованные комбинированные методы расчета [G4(MP2), G4] дают близкие результаты. Значения энтальпии образования, полученные на уровне расчета G4(MP2), превышают значения, полученные на уровне G4, на 76–118 кДж/кг, что составляет от 3 до 8 % для структур 1–10 и 24 % для структуры 11. При этом временные затраты на расчеты на уровне G4 в 3–5 раз больше, чем на уровне G4(MP2).

Таблица 1

Структуры, структурная формула, молекулярные веса (M_w , в а.е.), энтальпии образования (ΔH_f) рассчитанных молекул (в ккал/моль — обычный шрифт, *кДж/моль* — курсив, **кДж/кг** — полужирный шрифт), полученные на уровне G4(MP2) и G4

N M_w	Структурная формула	G4(MP2)	G4
1 $C_2N_6O_4$ 172.06		132.23 <i>553.27</i> 3215.55	128.76 <i>538.74</i> 3131.12
2 $C_2N_6O_5$ 188.06		125.25 <i>524.04</i> 2786.57	120.84 <i>505.59</i> 2688.49
3 $C_2N_6O_6$ 204.06		127.83 <i>534.84</i> 2621.04	122.72 <i>513.47</i> 2516.30
4 $C_2H_2N_6O_4$ 174.07		101.89 <i>426.29</i> 2448.86	97.67 <i>408.66</i> 2347.59
5 $C_3HN_7O_6$ 231.08		130.41 <i>545.64</i> 2361.20	124.95 <i>522.76</i> 2262.36
6 $C_2H_2N_6O_4$ 174.07		95.52 <i>399.64</i> 2295.80	91.53 <i>382.95</i> 2199.92
7 $C_3HN_7O_4F_2$ 237.08		122.83 <i>513.94</i> 2167.77	118.04 <i>493.86</i> 2083.10
8 $C_4N_{10}O_{12}$ 380.1		149.26 <i>624.57</i> 1643.16	138.54 <i>579.64</i> 1524.97
9 $C_3HN_6O_4F$ 204.06		74.92 <i>313.47</i> 1536.05	71.18 <i>297.81</i> 1459.29
10 $C_4N_{10}O_8F_4$ 392.1		133.75 <i>559.62</i> 1427.24	124.29 <i>520.05</i> 1326.33
11 $C_4N_8O_8F_2$ 326.1		37.96 <i>158.81</i> 487.02	30.63 <i>128.14</i> 392.97

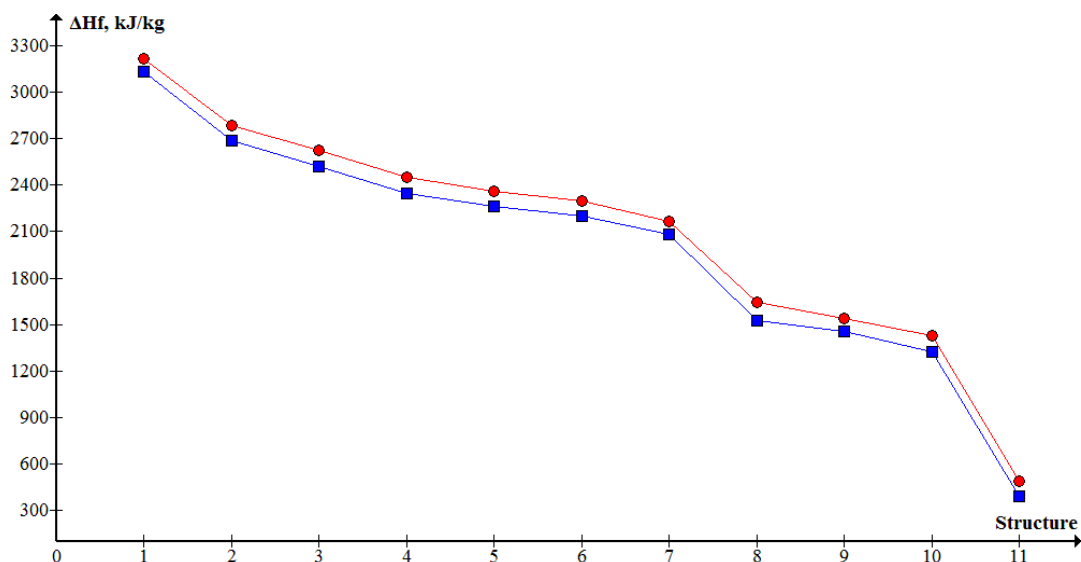


Рис. 2. Энтальпии образования структур 1–11 (ΔH_f , в кДж/кг).

Расчеты на уровне: G4(MP2) — красные круги, G4 — синие квадраты

Оттуда же видно, что рассмотренные в статье изменения в структуре **1** приводят к уменьшению энтальпии образования. Рассмотрим подробнее наблюдаемые в структурах **1–11** тенденции.

$C_2N_6O_4$. В ряду структур **1–3** смоделировано последовательное присоединение одного-двух атомов кислорода к кольцевой части структуры **1**. При этом энтальпия образования уменьшается последовательно на 429 и 594 кДж/кг, соответственно. При этом присоединение первого атома кислорода понижает энтальпию образования на 429, а второго — значительно меньше (на 166 кДж/кг).

Переход от структуры **3** к **4** путем замены одного из фрагментов NO_2 на NH_2 , приводит к уменьшению энтальпии образования на 172 кДж/кг.

В случае, когда атомы кислорода расположены противоположно друг другу (структура **4**), энтальпия образования больше (на 153 кДж/кг), чем для изомера со структурой **6**, где оба атома кислорода расположены вблизи NH_2 -фрагмента.

$C_3HN_7O_6$. Перестройка молекул в ряду от структуры **1** к структуре **5** осуществляется за счет замены одного фрагмента NO_2 на H, а другого — на $C(NO_2)_3$ и сопровождается уменьшением энтальпии образования на 854 кДж/кг. Замена в структуре **5** одного из фрагментов NO_2 на NF_2 (структура **7**) или F (структура **9**) приводит к уменьшению энтальпии образования на 193 кДж/кг и 825 кДж/кг, соответственно. Наличие двух противоположно расположенных групп $C(NO_2)_3$ в структуре **8** приводит к понижению энтальпии образования по сравнению со структурой **5** на 718 кДж/кг.

На рис. 3 приведены рассчитанные ИК-спектры поглощения исследуемых молекул. Смещения атомов для наиболее интенсивных колебаний приведены на рис. 4–14.

Наиболее высокие частоты (от 1628 см^{-1} до 1670 см^{-1}) с заметной интенсивностью имеют группы NO_2 , входящие в состав фрагментов R1 и R2 практически во всех рассматриваемых структурах. Исключение составляют структуры **4** и **6** (рис. 7 и рис. 9), в которых наибольшие по интенсивности частоты — это 1457 см^{-1} для структуры **4** и 1398 см^{-1} для структуры **6**, отвечающие колебаниям связи N-O атомов азота в кольце.

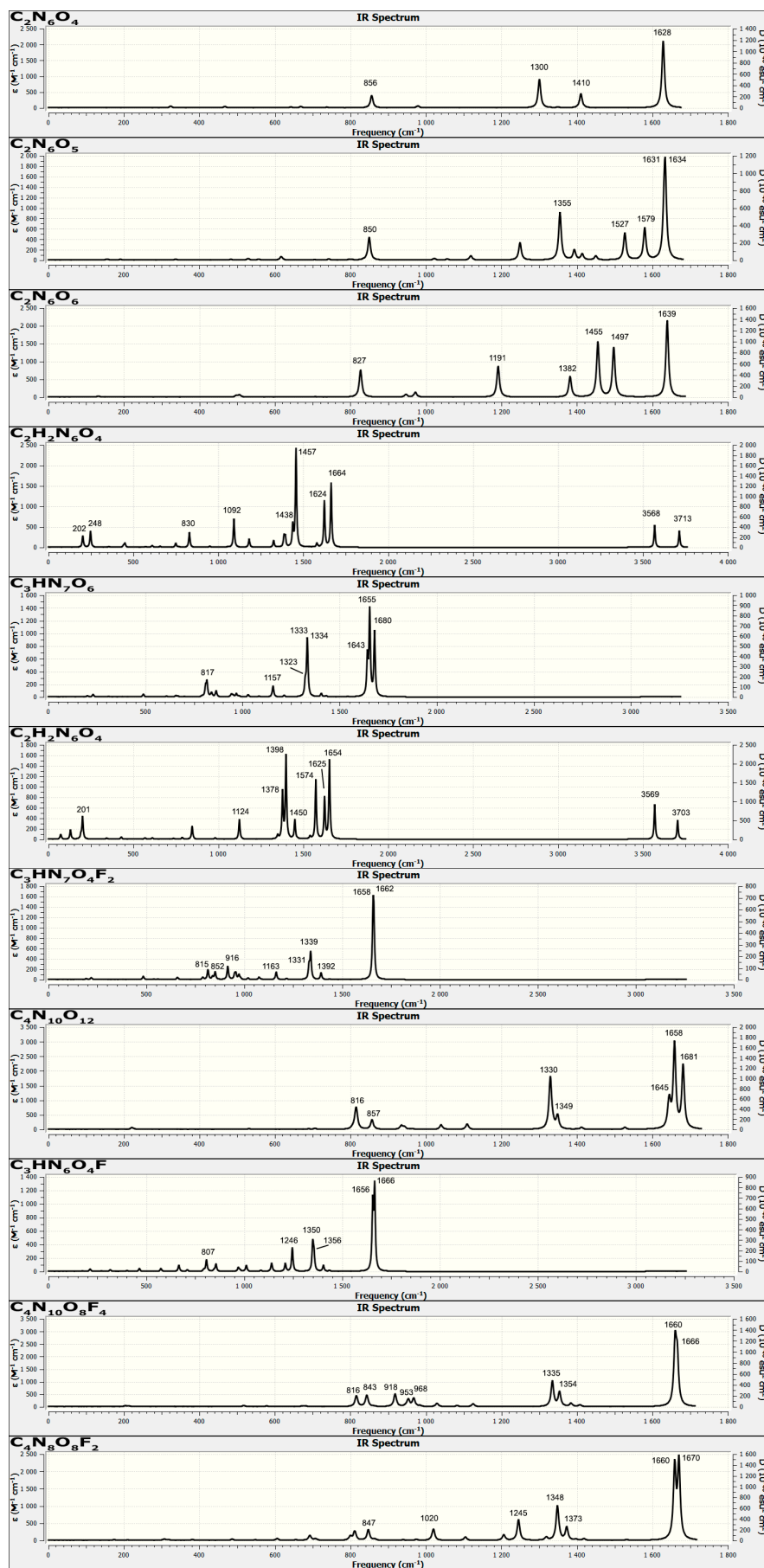


Рис. 3. ИК-спектры поглощения рассчитанных соединений

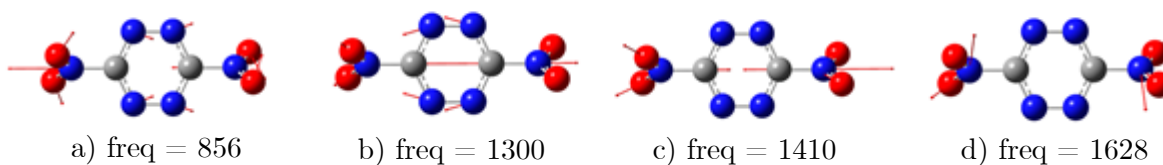


Рис. 4. Смещения атомов $C_2N_6O_4$ для указанных частот (структура 1)

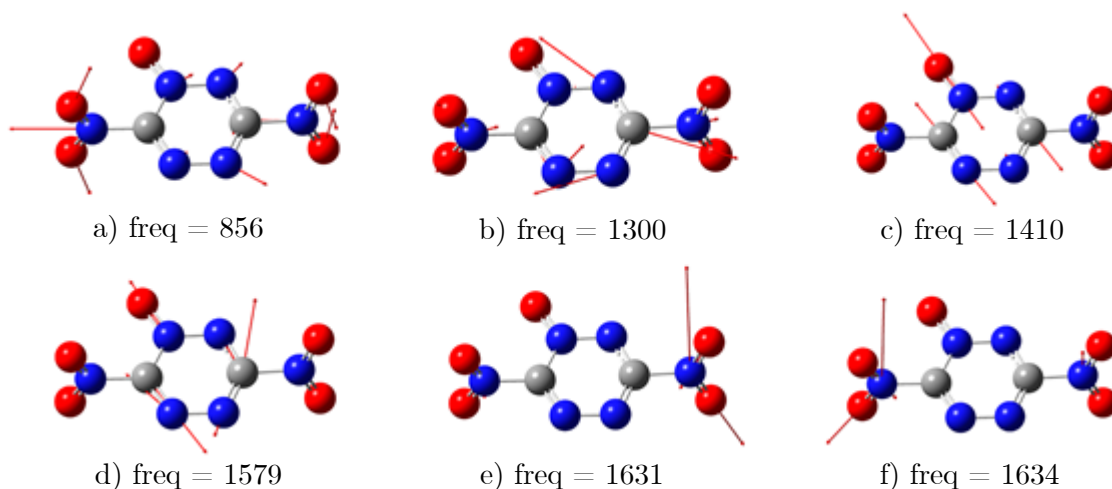


Рис. 5. Смещения атомов $C_2N_6O_5$ для указанных частот (структура 2)

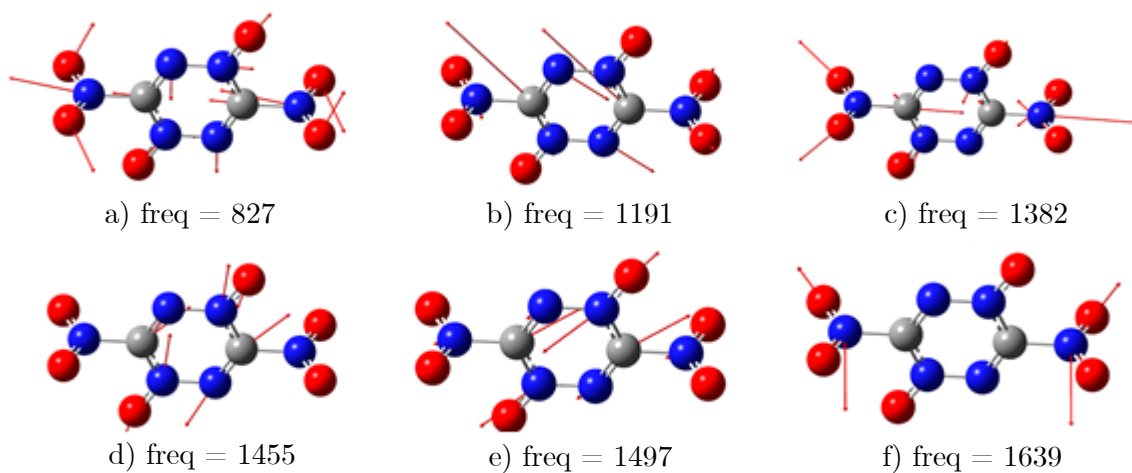


Рис. 6. Смещения атомов $C_2N_6O_6$ для указанных частот (структура 3)

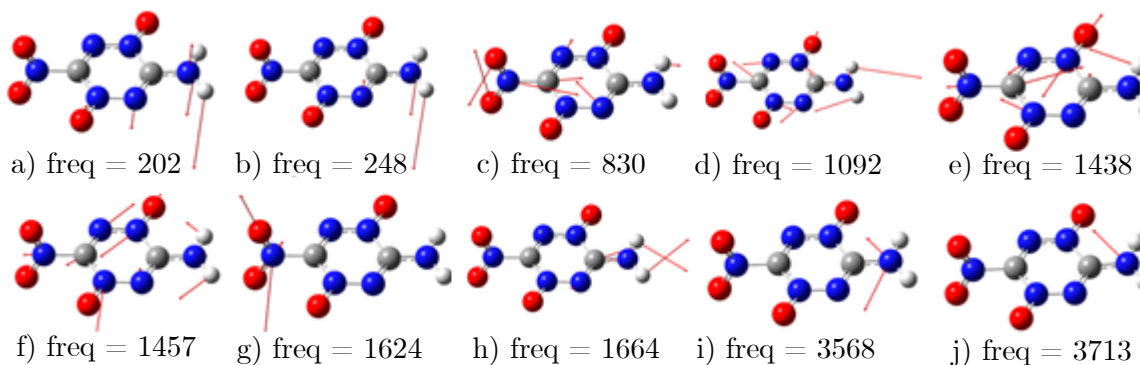


Рис. 7. Смещения атомов $C_2H_2N_6O_4$ для указанных частот (структура 4)

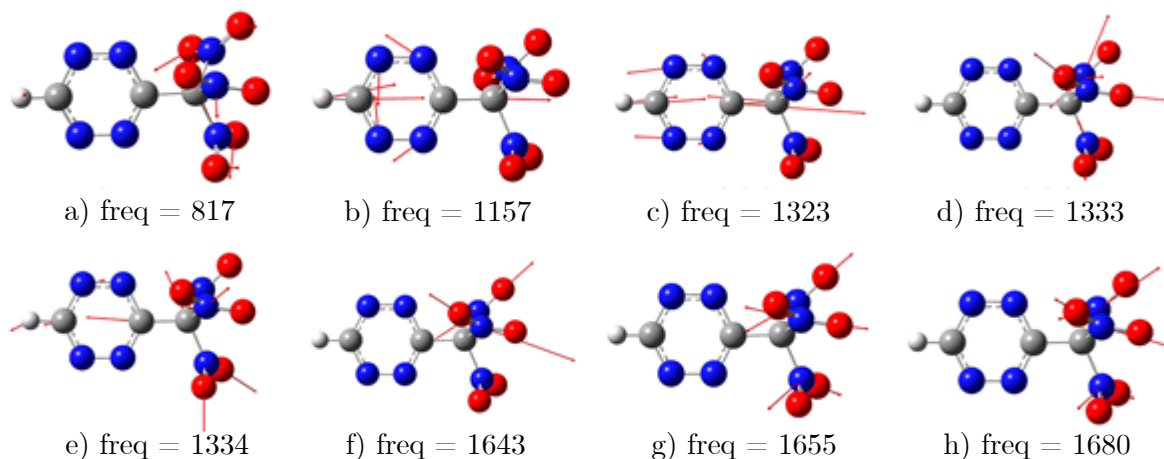


Рис. 8. Смещения атомов $C_3HN_7O_6$ для указанных частот (структура 5)

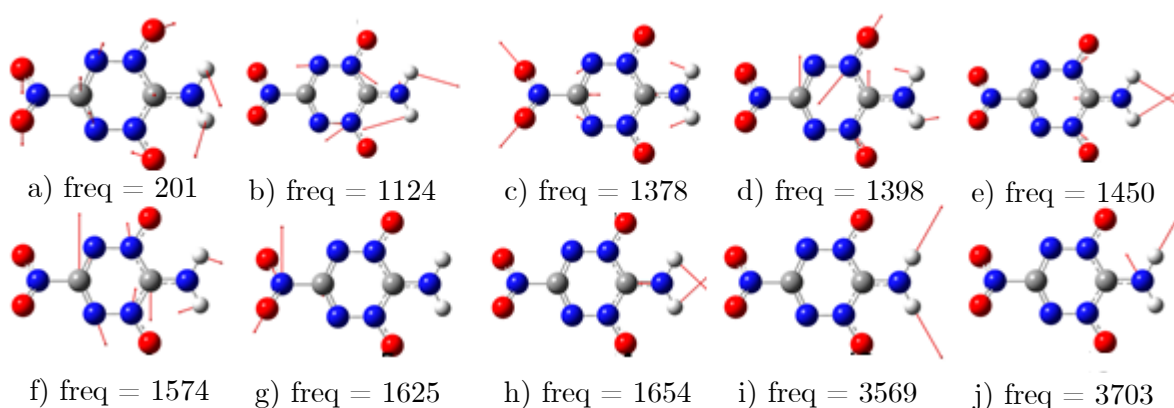


Рис. 9. Смещения атомов $C_2H_2N_6O_4$ для указанных частот (структура 6)

Колебания вдоль связи C-N атома азота, входящего в группу NO_2 , характеризуются частотами с меньшей интенсивностью: 1333 см^{-1} для структуры **5** (рис. 8), 1330 см^{-1} для структуры **8** (рис. 11), 1350 см^{-1} для структуры **9** (рис. 12), 1335 см^{-1} для структуры **10** (рис. 13) и 1348 см^{-1} для структуры **11** (рис. 14).

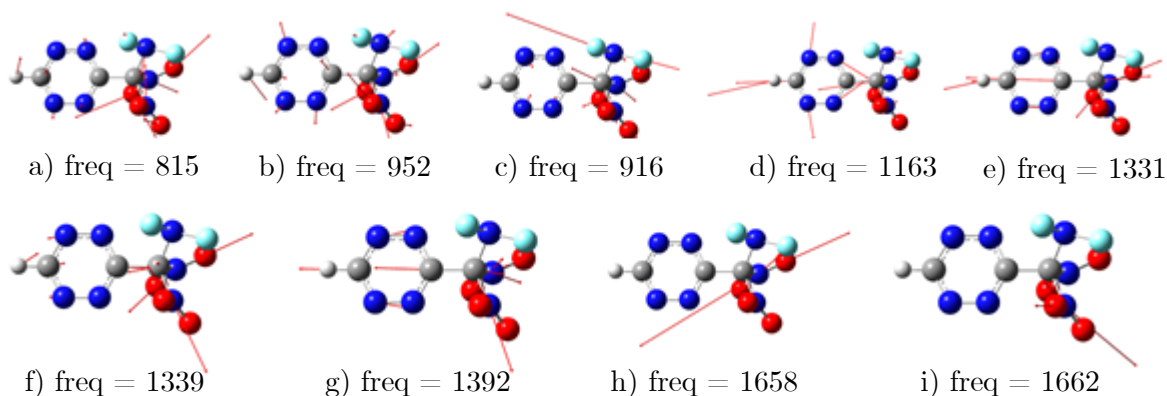


Рис. 10. Смещения атомов $C_3HN_7O_4F_2$ для указанных частот (структура 7)

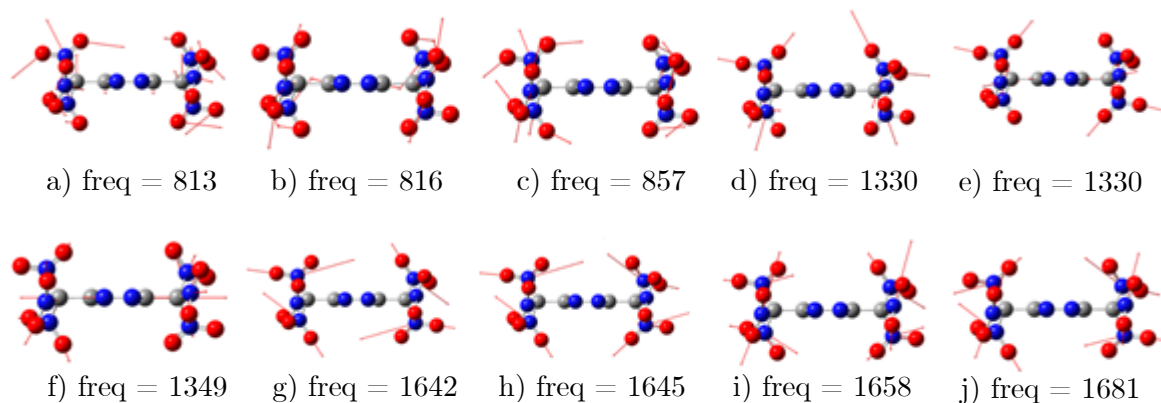


Рис. 11. Смещения атомов $C_4N_{10}O_{12}$ для указанных частот (структура 8)

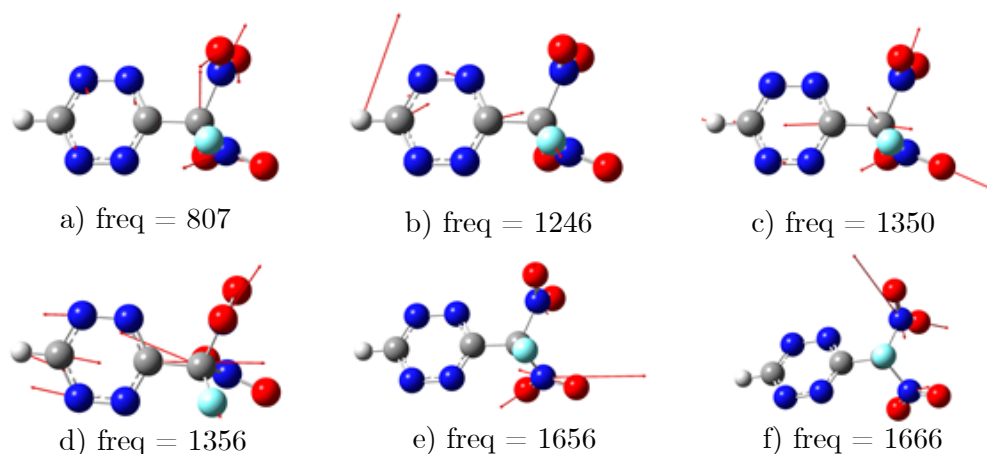


Рис. 12. Смещения атомов $C_3HN_6O_4F$ для указанных частот (структура 9)

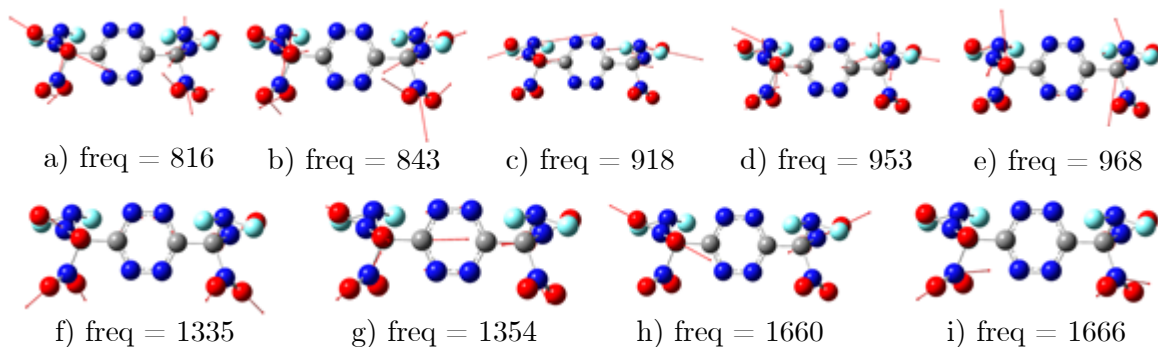


Рис. 13. Смещения атомов $C_4N_{10}O_8F_4$ для указанных частот (структура 10)

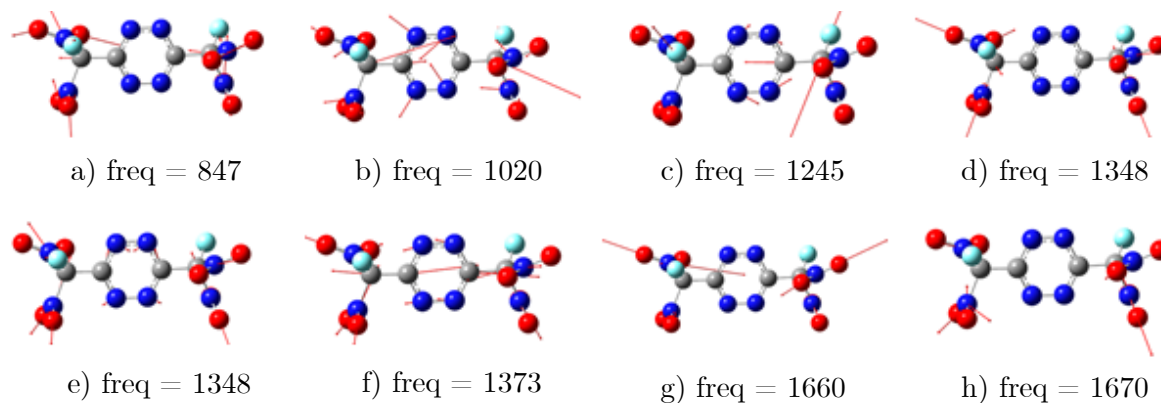


Рис. 14. Смещения атомов $C_4N_8O_8F_2$ для указанных частот (структура 11)

Заметной интенсивностью также характеризуются частоты, отвечающие асинхронным колебаниям атомов углерода и азота, входящих в кольцо: 1355 см⁻¹ для структуры **2** (рис. 5) и 1455 см⁻¹ для структуры **3** (рис. 6). Более подробная информация о наиболее интенсивных колебаниях приведена в табл. 2.

Таблица 2

Частоты и интенсивности колебаний

Frq.	Intens.	Frq.	Intens.	Frq.	Intensity	Frq.	Intensity	Frq.	Intensity	Frq.	Intensity
cm ⁻¹	km/mol	cm ⁻¹	km/mol	cm ⁻¹	km/mol	cm ⁻¹	km/mol	cm ⁻¹	km/mol	cm ⁻¹	km/mol
C₂N₆O₄		C₂N₆O₅		C₂N₆O₆		C₂H₂N₆O₄		C₃HN₇O₆		C₂H₂N₆O₄	
324	15.4	6167	19.9	506	17.7	202	83.5	809	45.7	72	27.9
856	113.1	850	127.1	827	221.9	248	116.3	817	66.7	130	52.2
979	18.9	1119	24.7	947	22.0	450	30.4	840	20.3	192	23.7
1300	261.1	1249	97.5	972	40.4	749	30.4	864	28.3	201	125.2
1410	130.5	1355	265.0	1191	252.4	830	107.1	968	16.1	846	73.7
1628	85.6	1393	55.3	1382	168.8	1092	200.7	1157	51.7	1124	111.1
1628	529.9	1414	33.5	1455	446.9	1181	59.5	1323	62.4	1349	19.8
		1450	22.5	1497	403.3	1325	49.2	1333	150.2	1378	259.3
		1527	150.2	1639	621.0	1387	74.7	1334	127.6	1398	460.4
		1579	177.3			1394	73.9	1405	16.4	1450	108.6
		1631	183.0			1438	152.5	1643	174.0	1540	16.5
		1634	444.3			1457	695.7	1655	387.2	1574	328.8
						1580	28.4	1680	293.4	1625	228.7
C₃HN₇O₄F₂		C₄N₁₀O₁₂		C₃HN₆O₄F		1624	329.6			1654	436.8
		221	16.8	666	27.1	1664	455.0			3569	191.8
484	20.1	814	101.3	807	50.9	3568	156.4			3703	106.3
814	56.5	816	115.7	855	32.2	3713	117.7	C₄N₈O₈F₂			
838	17.4	817	23.45	970	17.6			606	16.6		
852	43.4	857	95.8	1011	27.5	C₄N₁₀O₈F₄		693	42.6		
916	74.9	935	37.4	1140	37.3	814	35.7	800	33.8		
951	28.0	943	25.6	1210	34.8	816	95.2	811	50.8		
957	32.9	1040	46.3	1246	103.6	843	91.6	812	30.3		
973	27.7	1109	53.7	1350	118.0	844	43.4	847	90.1		
1163	43.6	1328	38.2	1356	71.1	918	143.9	1020	87.5		
1331	76.3	1328	40.7	1404	27.1	953	84.9	1104	30.3		
1339	146.6	1330	239.6	1656	276.7	968	81.3	1206	47.2		
1392	38.5	1330	210.9	1666	350.2	1029	39.0	1245	174.7		
1658	318.9	1337	27.0			1125	36.0	1319	26.0		
1662	276.5	1349	132.9			1335	146.8	1348	127.3		
		1412	20.2			1335	146.6	1348	138.4		
		1527	20.2			1354	166.8	1351	38.3		
		1643	138.5			1384	38.0	1373	109.3		
		1645	169.0			1408	21.3	1659	569.5		
		1658	804.8			1660	608.0	1660	42.9		
		1660	37.6			1660	99.8	1668	21.3		
		1681	620.4			1666	519.3	1670	630.2		

3. Особенности расчетов

При проведении квантово-химического моделирования был использован ряд вычислительных конфигураций (табл. 3) на базе различных модификаций процессоров Intel Xeon, предоставляемых пулов расчетных ядер, оперативной и дисковой памяти, наличия GPU, а также версий прикладного пакета Gaussian (<https://gaussian.com>).

Вычислительная сложность задач достаточно высока, для указанных структур в зависимости от использованного базиса и расчетных температур среднее время расчета варьирует от 20 часов до 30 дней (на конфигурации ИПХФ РАН, см. ниже), при использовании наиболее современных версий процессоров время расчетов падает, однако, для более сложных систем (до 30 атомов, например для $C_4N_{10}O_8F_4$) время расчетов на 24 ядерной рабочей станции в псевдооднозадачном режиме все же достигало 2.5 месяцев.

Критично к скорости расчетов поддержка используемыми процессорами инструкций **avx2** и **sse42**, особенно первой — на некоторых задачах при близкой тактовой частоте процессоров выигрыш может составлять 8–10 раз [16]. К сожалению, нам не удалось оценить возможности использования GPU ускорителей, т.к. Gaussian использует только графические ускорители серий K40, K80 и P100, которых не было в нашем распоряжении. Более старые ускорители не поддерживаются.

Таблица 3

Аппаратная платформа вычислительные конфигурации

Вычислительный ресурс	Процессоры/Число ядер/ RAM/GPU	Использование
Ломоносов-2«compute»	Intel Xeon® E5-2697 v3@2.60 ГГц, 14 ядер, 64 Гб, Tesla K40s	до 104 ядер, 64 Гб
Вычислительный кластер ИПХФ РАН	Intel Xeon® 5450 5670@ 3 ГГц, 4–6 ядер, 8 и 12 Гб RAM	от 1 до 50 CPU (до 200 ядер), от 1 до 12 Гб на узел
Рабочие станции ИПХФ РАН с GPU Tesla C2075	Intel Xeon® X5675@3.46 ГГц, 2x6 ядер, 48 Гб RAM, GPU Nvidia Tesla C2075	до 12 ядер
Рабочая станция ИЭМ РАН	Intel Xeon® E5-2690v3, 2x12 ядер, RAM 256 Гб, SSD+4 Тб HDD	1–24 ядра, 4–248 Гб RAM на задачу

примечание: приведено количество физических ядер без потоков (threads)

Крайне желательно проведение расчетов с использованием SSD или высокоскоростных SAS дисков с большим объемом выделяемой дисковой памяти, поскольку пакет создает в процессе вычислений гигантские промежуточные файлы до 2 Тб размером, запись которых может занимать до 30–35 минут на SSD диске и, естественно, существенно дольше на SATA массивах

Сильно влияет на скорость расчетов наличие последних версий пакета Gaussian по сравнению с g9 (установленной на кластере ИПХФ), которые в полной мере реализуют аппаратные возможности процессоров новых серий, давая ускорение расчетов для большинства использованных базисов до 7–8 раз. К сожалению, зачастую последние версии пакета даже не запускаются на процессорах старше 5–6 лет.

Перед авторами не ставилась задача провести детальный анализ степени параллелизации выполняемых расчетов (ввиду того, что пакет Gaussian использует свою собственную внутреннюю систему распараллеливания Linda), однако обычно наблюдается устойчивое ускорение на пулах до 12 ядер, далее эффект снижается. Это также зависит от объема выделяемой задаче памяти (при этом требуется не менее 4 Гб RAM на физическое ядро).

Заключение

В результате проведения высокоточных квантово-химических расчетов получены термодинамические данные высокоэнергетических соединений: $C_2N_6O_4$, $C_2N_6O_5$, $C_2N_6O_6$, $C_2H_2N_6O_4$, $C_3HN_7O_6$, $C_3HN_7O_4F_2$, $C_4N_{10}O_{12}$, $C_3HN_6O_4F$, $C_4N_{10}O_8F_4$, $C_4N_8O_8F_2$. Показано, что энтальпия образования уменьшается с усложнением строения молекул исследуемого ряда. Рассчитаны также ИК-спектры поглощения, структурные параметры и смещения атомов для наиболее интенсивных колебаний. Расчеты выполнены с использованием комбинированных методов G4(MP2) и G4. Продемонстрировано, что использование уровня расчета G4(MP2) для выбранного класса молекул дает близкие (в пределах 8 %) относительно уровня G4 результаты и ведет к существенной (в 3–5 раз) экономии расчетного времени.

Работа выполнена с использованием оборудования Центра коллективного пользования высокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова (проект Enthalpy-2065) [17–19] и собственных вычислительных ресурсов ИПХФ РАН. Авторы также приносят благодарности группе компаний РСК и Институту экспериментальной минералогии РАН за предоставленные для части расчетов пулы вычислительных ресурсов. Работа выполнена по теме Государственного задания, № государственной регистрации 00892019-0017 [АААА-А19-119120690042-9], а также при поддержке РФФИ (проект №20-07-00319).

Литература

1. Hosseini S.G., Moeini K., Abdelbaky M.S.M., Garcia-Granda S. Synthesis, characterization, crystal structure, and thermal behavior of a new triazolium salt along with docking studies // J. Struct. Chem. 2020. Vol. 61, no. 3. P. 389–399. DOI: 10.26902/JSC_id52850.
2. Абдулов Х.Ш., Муллоев Н.У. Табаров С.Х., Ходиев М.Х. Квантово-химическое определение молекулярной структуры 1,2,4-триазола и расчет его инфракрасного спектра // Журнал структурной химии. 2020. Т. 61, № 4. С. 540–544. DOI: 10.26902/JSC_id53992.
3. Lv G., Zhang D.-L., Wang D., Pan L., Liu Y. Synthesis, crystal structure, anti-bone cancer activity and molecular docking investigations of the heterocyclic compound 1-((2S,3S)-2-(benzyloxy)pentan-3-yl)-4-(4-(4-(4-hydroxyphenyl)piperazin-1-yl) phenyl)-1H-1,2,4-triazol-5(4H)-one // J. Struct. Chem. 2019. Vol. 60, no. 7. P. 1219–1225. DOI: 10.26902/JSC_id43057.
4. Волохов В.М., Зюбина Т.С., Волохов А.В. и др. Квантово-химическое моделирование углеводородных соединений с высокой энтальпией образования // Химическая физика. 2021. Т. 40, № 1. С. 3–15. DOI: 10.31857/S0207401X21010131.
5. Волохов В.М., Зюбина Т.С., Волохов А.В. и др. Предсказательное моделирование молекул высокоэнергетических гетероциклических веществ // Журнал неорганической химии. 2021. Т. 66, № 1. С. 69–80. DOI: 10.31857/S0044457X21010116.
6. Frisch M.J., Trucks G.W., Schlegel H.B. et al. Gaussian 09, Revision B.01. Gaussian Inc., Wallingford CT, 2010. 487 p.
7. Becke A.D. Density functional calculations of molecular bond energies // J. Chem. Phys. 1986. Vol. 84. P. 4524. DOI: 10.1063/1.450025.
8. Johnson B.J., Gill P.M.W., Pople J.A. The performance of a family of density functional methods // J. Chem. Phys. 1993. Vol. 98, no. 4. P. 5612. DOI: 10.1063/1.464906.

9. Curtiss L.A. Gaussian-4 theory // J. Chem. Phys. 2007. Vol. 126, no. 8. P. 084108. DOI: 10.1063/1.2436888.
10. Curtiss L.A., Redfern P.C., Raghavachari K. G4 theory // Comput. Mol. Sci. 2011. Vol. 1. P. 810–825. DOI: 10.1002/wcms.59.
11. Curtiss L.A., Raghavachari K., Redfern P.C., Pople J.A. Assessment of Gaussian-2 and density functional theories for the computation of enthalpies of formation // J. Chem. Phys. 1997. Vol. 106, no. 3. P. 1063. DOI: 10.1063/1.473182.
12. NIST-JANAF Thermochemical tables. URL: <https://janaf.nist.gov/> (дата обращения: 27.09.2020).
13. Computational Chemistry Comparison and Benchmark DataBase. URL: <https://cccbdb.nist.gov/hf0k.asp> (дата обращения: 10.10.2020).
14. Ефимов А.И., Белорукова Л.П., Василькова И.В. и др. Свойства неорганических соединений. Справочник. Л.: Химия, 1983. 392 с.
15. Гурвич Л.В. Энергии разрыва химических связей. М.: Наука, 1974. 351 с.
16. Grigorenko V., Mironov V., Polyakov I., Nemukhin A. Benchmarking quantum chemistry methods in calculations of electronic excitations // Supercomput. Front. Innov. 2019. Vol. 5, no. 4. P. 62–66. DOI: 10.14529/jsfi180405.
17. Voevodin V.I. et al. Supercomputer Lomonosov-2: large scale, deep monitoring and fine analytics for the user community // Supercomput. Front. Innov. 2019. Vol. 6, no. 2. P. 4–11. DOI: 10.14529/jsfi190201.
18. Воеводин В., Жуматий С., Соболев С., Антонов А., Брызгалов П., Никитенко Д., Стефанов К., Воеводин Вад. Практика суперкомпьютера «Ломоносов» // Открытые системы. 2012. Т. 7. С. 36–39.
19. Nikitenko D., Voevodin V., Zhumatiy S. Deep analysis of job state statistics on “Lomonosov-2” supercomputer // Supercomput. Front. Innov. 2019. Vol. 5, no. 2. P. 4–10. DOI: 10.14529/jsfi180201.

Волохов Вадим Маркович, д.ф.-м.н., профессор, г.н.с., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Амосова Елена Сергеевна, м.н.с., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Волохов Александр Вадимович, н.с., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Зюбина Татьяна Сергеевна, д.х.н., в.н.с., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Лемперт Давид Борисович, к.х.н., зав. лаб., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Яновский Леонид Самойлович, д.т.н., профессор, зав. лаб., Институт проблем химической физики РАН (Черноголовка, Российская Федерация)

Варламов Дмитрий Анатольевич, с.н.с., Институт проблем химической физики РАН (Черноголовка, Россия)

QUANTUM-CHEMICAL MODELING OF THERMOCHEMICAL PROPERTIES OF MOLECULES OF HIGH-ENERGY TETRAZINES

© 2021 V.M. Volokhov¹, E.S. Amosova¹, A.V. Volokhov¹, T.S. Zyubina¹,
D.B. Lempert¹, L.S. Yanovskiy^{1,2,3}, D.A. Varlamov¹

¹*Institute of Problems of Chemical Physics of the Russian Academy of Sciences
(av. acaf. Semenov 1, Chernogolovka, Moscow Region, 142432 Russia),*

²*The P.I. Baranov Central Institute of Aviation Motor Development
(str. Aviamotornaya 2, Moscow, 111116 Russia),*

³*Moscow Aviation Institute (National Research University)
(av. Volokolamskoye 4, Moscow, 125993 Russia)*

E-mail: {vvm, aes, vav, zyubin, lempert, Yanovskiy, dima}@icp.ac.ru

Received: 11.05.2021

The article presents the results of calculations of structural and thermochemical characteristics of a number of potentially high-energy compounds $C_2N_6O_4$, $C_2N_6O_5$, $C_2N_6O_6$, $C_2H_2N_6O_4$, $C_3HN_7O_6$, $C_3HN_7O_4F_2$, $C_4N_{10}O_{12}$, $C_3HN_6O_4F$, $C_4N_{10}O_8F_4$, $C_4N_8O_8F_2$ using quantum chemical *ab initio* methods. Quantum-chemical modeling has been performed using the GAUSSIAN 09 applied software package. Using the B3LYP hybrid density functional, the search for the optimal molecular geometry has been carried out, the IR absorption spectra, structural parameters and atomic displacements for the most intense vibrations have been calculated. The enthalpies of formation (EO) have been obtained by the atomization method and are provided in the article. The calculations have been performed using the combined G4(MP2) and G4 methods within the Gaussian 09 application package. It is shown that the enthalpy of formation depends on the molecule structure. Examples of the used computational configurations based on several computational resources are given, some features of the calculations are described. Comparison of time costs and errors is made when using different methods on the same structures.

Keywords: quantum-chemical calculations, Gaussian package, enthalpy of formation, high-enthalpy compounds, IR spectra of gaseous molecules, combined G4 and G4(MP2) methods.

FOR CITATION

Volokhov V.M., Amosova E.S., Volokhov A.V., Zyubina T.S., Lempert D.B., Yanovskiy L.S., Varlamov D.A. Quantum-chemical Modeling of Thermochemical Properties of Molecules of High-energy Tetrazines. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 82–96. (in Russian) DOI: 10.14529/cmse210206.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Hosseini S.G., Moeini K., Abdelbaky M.S.M., Garcia-Granda S. Synthesis, characterization, crystal structure, and thermal behavior of a new triazolium salt along with docking studies. *J. Struct. Chem.* 2020. Vol. 61, no. 3. P. 389–399. DOI: 10.26902/JSC_id52850.
2. Abdulov Kh.Sh., Mulloev N.U., Tabarov Kh.S., Khodiev M.Kh. Quantum Chemical Determination of the Molecular Structure of 1,2,4-Triazole and the Calculation of its Infrared Spectrum. *J. Struct. Chem.* 2020. Vol. 61, no. 4. P. 510–514. DOI: 10.1134/S0022476620040022.

3. Lv G., Zhang D.-L., Wang D., Pan L., Liu Y. Synthesis, crystal structure, anti-bone cancer activity and molecular docking investigations of the heterocyclic compound 1-((2S,3S)-2-(benzyloxy)pentan-3-yl)-4-(4-(4-(4-hydroxyphenyl)piperazin-1-yl) phenyl)-1H-1,2,4-triazol-5(4H)-one. *J. Struct. Chem.* 2019. Vol. 60, no. 7. P. 1219–1225. DOI: 10.26902/JSC_id43057.
4. Volokhov V.M., Zyubina T.S., Volokhov A.V., Amosova E.S., Varlamov D.A., Lempert D.B., Yanovskii L.S. Quantum Chemical Simulation of Hydrocarbon Compounds with High Enthalpy. *Russian Journal of Physical Chemistry B.* 2021. Vol. 15, no. 1. P. 12–24. DOI: 10.1134/S1990793121010127.
5. Volokhov V.M., Zyubina T.S., Volokhov A.V., Amosova E.S., Varlamov D.A., Lempert D.B., Yanovskii L.S. Predictive Modeling of Molecules of High-Energy Heterocyclic Compounds. *Russian Journal of Inorganic Chemistry.* 2021. Vol. 66, no. 1. P. 78–88. DOI: 10.1134/S0036023621010113.
6. Frisch M.J., Trucks G.W., Schlegel H.B. et al. Gaussian 09, Revision B.01. Gaussian, Inc., Wallingford CT, 2010. 487 p.
7. Becke A.D. Density functional calculations of molecular bond energies. *J. Chem. Phys.* 1986. Vol. 84. P. 4524 DOI: 10.1063/1.450025.
8. Johnson B.J., Gill P.M.W., Pople J.A. The performance of a family of density functional methods. *J. Chem. Phys.* 1993. Vol. 98, no. 4. P. 5612. DOI: 10.1063/1.464906.
9. Curtiss L.A. Gaussian-4 theory. *J. Chem. Phys.* 2007. Vol. 126, no. 8. P. 084108. DOI: 10.1063/1.2436888.
10. Curtiss L.A., Redfern P.C., Raghavachari K. G3 theory. *WIREs Comput. Mol. Sci.* 2011. Vol. 1. P. 810–825. DOI: 10.1002/wcms.59.
11. Curtiss L.A., Raghavachari K., Redfern P.C., Pople J.A. Assessment of Gaussian-2 and density functional theories for the computation of enthalpies of formation. *J. Chem. Phys.* 1997. Vol. 106, no. 3. P. 1063. DOI: 10.1063/1.473182.
12. NIST-JANAF Thermochemical tables URL: <https://janaf.nist.gov> (accessed: 27.09.2020)
13. Computational Chemistry Comparison and Benchmark DataBase URL: <https://cccbdb.nist.gov/hf0k.asp> (accessed: 10.10.2020).
14. Efimov A.I., Belorukova L.P., Vasilkova I.V. et al. Properties of inorganic compounds. Reference book. Khimiya, Leningrad. 1983. 392 p. (in Russian)
15. Gurvich L.V. Bond Dissociation Energies. Nauka, Moscow. 1974. 351 p. (in Russian)
16. Grigorenko B., Mironov V., Polyakov I., Nemukhin A. Benchmarking quantum chemistry methods in calculations of electronic excitations. *Supercomput. Front. Innov.* 2019. Vol. 5, no. 4. P. 62–66. DOI: 10.14529/jsfi180405.
17. Voevodin V.I., Antonov A.S., Nikitenko D.A., Shvets P.A., Sobolev S.I., Sidorov I.Yu., Stefanov K.S., Voevodin V.V., Zhumatiy S.A. Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community. *Supercomput. Front. Innov.* 2019. Vol. 6, no. 2. P. 4–11. DOI: 10.14529/jsfi190201.
18. Voevodin V.V. et al. Practice of “Lomonosov” supercomputer. *Open Systems.* 2012. Vol. 7. P. 36–39. (in Russian)
19. Nikitenko D., Voevodin V., Zhumatiy S. Deep analysis of job state statistics on “Lomonosov-2” supercomputer. *Supercomput. Front. Innov.* 2019. Vol. 5, no. 2. P. 4–10. DOI: 10.14529/jsfi180201.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки), 05.13.17 – Теоретические основы информатики (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции, издателя: 454080, г. Челябинск, проспект Ленина, 76, Издательский центр ЮУрГУ, каб. 32.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии <http://vestnikvmi.susu.ru>. Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:
Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, кафедра СП,
ответственному секретарю Цымблеру М.Л.
3. Адрес электронной почты редакции: vestnikvmi@susu.ru
4. Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.

ВЕСТНИК
ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
Серия
«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»
Том 10, № 2
2021

16+

Техн. редактор *А.В. Миних*

Издательский центр Южно-Уральского государственного университета

Подписано в печать 31.05.2021. Дата выхода в свет 07.06.2021. Формат 60×84 1/8. Печать цифровая.
Усл. печ. л. 11,62. Тираж 500 экз. Заказ 135/213. Цена свободная.

Отпечатано в типографии Издательского центра ЮУрГУ.
454080, г. Челябинск, проспект Ленина, 76.