

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 27

2021

№ 11

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

Рисунки к статье А. А. Сорокина

«ИССЛЕДОВАНИЕ ИЕРАРХИЧЕСКИХ СИСТЕМ НЕЧЕТКОГО ВЫВОДА
ПРИ ПОЛУЧЕНИИ ИНТЕГРАЛЬНЫХ ОЦЕНОК АНАЛИЗИРУЕМЫХ ОБЪЕКТОВ»

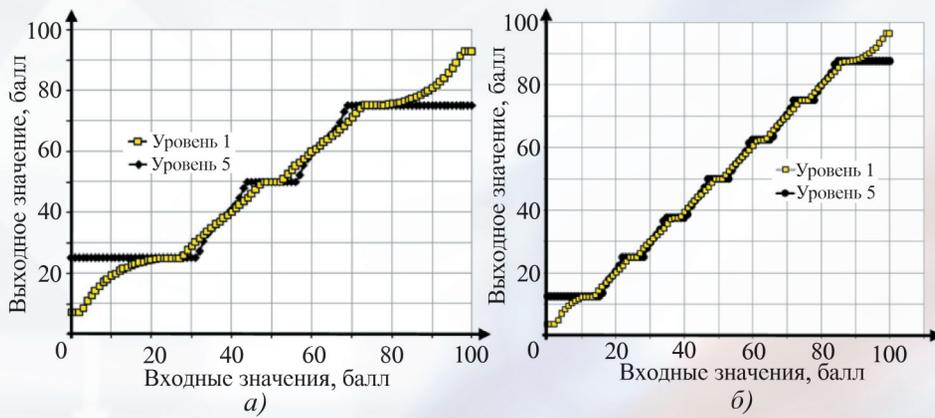


Рис. 3. Результат 1-го цикла моделирования для ИСНВ-М

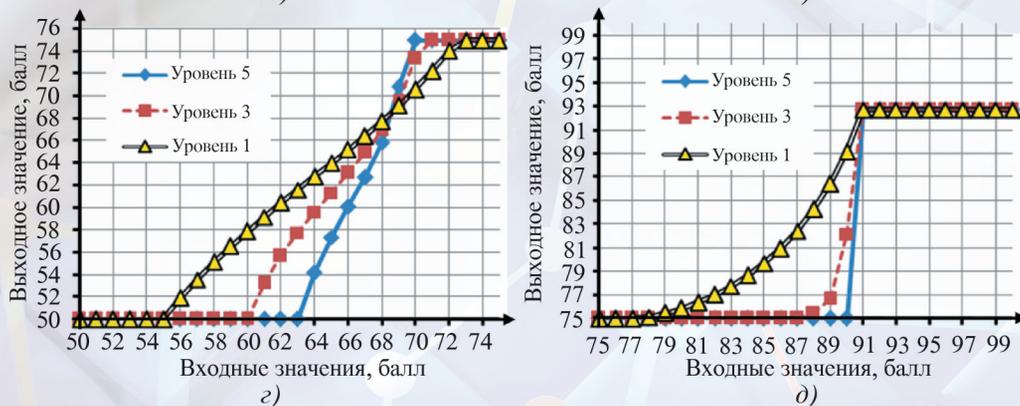
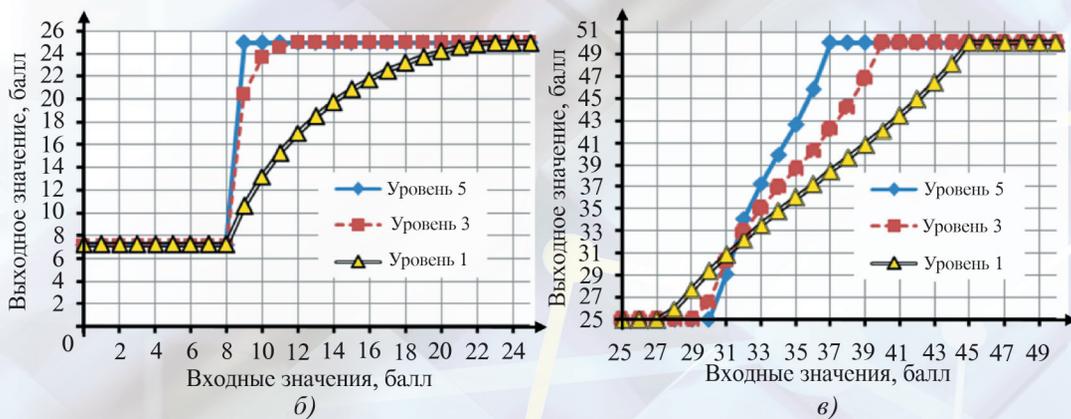
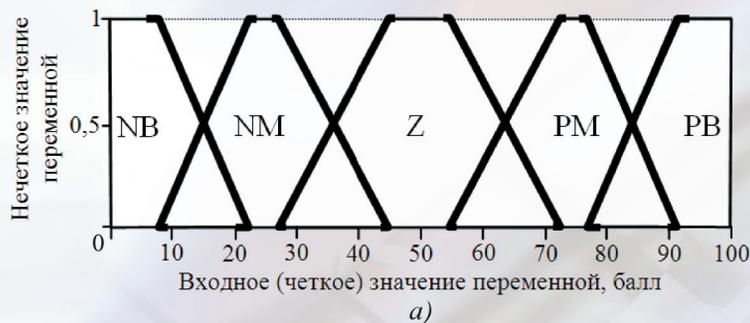


Рис. 7. Результат моделирования ИСНВ-М после корректировки термов

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 27
2021
№ 11

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

Курейчик В. В., Родзин С. И. Вычислительные модели биоэвристик, основанных на физических и когнитивных процессах (обзор) 563

Гридин В. Н., Анисимов В. И., Васильев С. А. Методы расчета переходных процессов в больших электронных схемах на основе диакоптики 575

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Сорокин А. А. Исследование иерархических систем нечеткого вывода при получении интегральных оценок анализируемых объектов 582

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

Асратян Р. Э. Сетевая служба параллельной обработки защищенных информационных запросов в мультисерверной среде 592

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Осипович В. С., Медведев О. С., Дудич О. Н., Красильникова В. Л., Яшин К. Д. Технология преобразования результатов компьютерной томографии в трехмерные модели 600

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Гриняк В. М., Шуленина А. В. Кластеризация данных траекторий морских судов для планирования маршрутов через акватории с интенсивным движением 607

Главный редактор:

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

Зам. главного редактора:

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

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.

ЖУРАВЛЕВ Ю. И.,

акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,

акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,

акад. РАН, д. т. н., проф.

РУСАКОВ С. Г.,

чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,

чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,

акад. РАН, д. т. н., проф.

СОКОЛОВ И. А.,

акад. РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.

ЧАПЛЫГИН Ю. А.,

акад. РАН, д. т. н., проф.

ШАХНОВ В. А.,

чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,

акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,

чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.

АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.

ВАСЕНИН В. А., д. ф.-м. н., проф.

ВАСИЛЬЕВ В. И., д. т. н., проф.

ВИШНЕКОВ А. В., д. т. н., проф.

ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В. Г., д. т. н., проф.

ЗАБОРОВСКИЙ В. С., д. т. н., проф.

ЗАРУБИН В. С., д. т. н., проф.

КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.

КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. В., д. т. н., проф.

ЛЬВОВИЧ Я. Е., д. т. н., проф.

МАРТЫНОВ В. В., д. т. н., проф.

МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.

ПОЛЕЩУК О. М., д. т. н., проф.

ПРОХОРОВ С. А., д. т. н., проф.

САКСОНОВ Е. А., д. т. н., проф.

СОКОЛОВ Б. В., д. т. н., проф.

СОЛОВЬЕВ Р. А., д. т. н., в. н. с.

ТИМОНИНА Е. Е., д. т. н., проф.

УСКОВ В. Л., к. т. н. (США)

ФИНЬКО О. А., д. т. н., проф.

ФОМИЧЕВ В. А., д. т. н., проф.

ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 27
2021
No. 11

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

DOI 10.17587/issn.1684-6400

ISSN 1684-6400

CONTENTS

MODELING AND OPTIMIZATION

- Kureychik V. V., Rodzin S. I.** Computational Models of Bio Heuristics Based on Physical and Cognitive Processes (Review) 563
- Gridin V. N., Anisimov V. I., Vasilev S. A.** Methods for Calculating Transients in Large Electronic Circuits Based on Diacoptics 575

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Sorokin A. A.** Investigation of Hierarchical Fuzzy Inference Systems, when Obtaining Integral Estimates of the Analyzed Objects 582

INFORMATION SECURITY

- Asratian R. E.** A Service for Parallel Processing of Protected Information Queries in a Multi-Server Environment 592

DIGITAL PROCESSING OF SIGNALS AND IMAGES

- Osipovich V. S., Medvedev O. S., Dudzich O. N., Krasilnikova V. L., Yashin K. D.** Technology for Conversion of Computer Tomography Results into Three-Dimensional Models 600

APPLICATION INFORMATION SYSTEMS

- Grinyak V. M., Shulenina A. V.** Marine Traffic Data Clustering for Ships Route Planning 607

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

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

Editors:

Bezmenova M. Yu.

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

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

В. В. Курейчик, д-р техн. наук, проф., e-mail: vkur@sfedu.ru,
С. И. Родзин, канд. техн. наук, проф., e-mail: srodzin@yandex.ru,
Южный федеральный университет, г. Таганрог

Вычислительные модели биоэвристик, основанных на физических и когнитивных процессах (обзор)^{1, 2}

Приводятся вычислительные модели биоэвристик, основанных на физических и когнитивных процессах. Со- поставляются данные о таких характеристиках биоэвристик (включая эволюционные и роевые биоэвристики), как скорость сходимости, вычислительная трудоемкость, требуемый объем памяти, настройка параметров алгоритма, трудности программной реализации. Оценивается баланс между скоростью сходимости биоэвристик и диверсификацией пространства поиска решений оптимизационных задач. Представлены экспериментальные результаты для задачи размещения графов Ресо в решетке с минимальной суммарной длиной ребер графа.

Ключевые слова: биоэвристика, сходимость алгоритма, диверсификация пространства поиска, популяция, фитнес-функция, оптимизация, граф

Введение

Биоэвристики лучше всего работают над широким спектром оптимизационных задач, когда в их механизме присутствует баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений. Диверсификация пространства поиска решений относится к способности поисковых агентов посещать новые регионы пространства поиска, в то время как при анализе сходимости алгоритма делается акцент на возможностях этих агентов уточнять найденные "хорошие" решения. Поиск такого баланса является фундаментальной задачей. В существующих биоэвристиках он нередко слабо выражен, к тому же отсутствуют инструменты измерения этого баланса, позволяющие понять, как на алгоритм влияют механизмы поиска оптимального решения. Еще один фактор, который сильно влияет на баланс, — выбор параметров (настройка) алгоритма, предназначенных для управления скоростью сходимости

алгоритма и диверсификацией пространства поиска решений. Процесс выбора параметров не является простым и зависит от решаемой задачи.

В данной второй части обзора представлены особенности биоэвристик, основанных на физических процессах, и биоэвристик, основанных на когнитивных процессах и деятельности человека. Также приводится сравнительный анализ эволюционных и роевых биоэвристик, рассмотренных в первой части обзора, и биоэвристик, основанных на физических процессах, на когнитивных процессах и деятельности человека.

1. Биоэвристики, основанные на физических процессах

Биоэвристики, основанные на физических процессах, включают алгоритмы движения заряженных частиц (*EM*), гравитационного взаимодействия (*GSA*), моделирования отжига (*SA*), синусоидальных функций (*SCA*), перехода вещества из одного теплового состояния в другое (*SMS*), каплей воды (*ADW*).

Алгоритм движения заряженных частиц (EM), предложенный С. Бирбиллом и Ш. Фангом в 2003 г. [1], инспирирован физическими законами, которые управляют движением заряженных частиц в заданном пространстве.

¹Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00570.

²Данный обзор является продолжением статьи "Вычислительные модели эволюционных и роевых биоэвристик (обзор)", опубликованной в журнале "Информационные технологии", 2021, Т.27, № 10.

В *EM* поисковыми агентами являются взаимодействующие в пространстве электрически заряженные частицы. Каждой i -й частице присваивается индивидуальное значение заряда согласно следующему выражению:

$$q_i^k = \exp \left(-d \frac{f(x_i^k) - f(x_{best}^k)}{\sum_{j=1}^N (f(x_j^k) - f(x_{best}^k))} \right), \quad (1)$$

где x_{best}^k — текущее лучшее решение, найденное до сих пор алгоритмом, d — размерность пространства решений.

Каждая из частиц согласно закону Кулона подвергается воздействию электростатических сил притяжения либо отталкивания, зависящих от величины и знака их зарядов, а также расстояния между ними. Общая электростатическая сила i -й частицы определяется как

$$F_i^k = \sum_{j \neq i}^N \begin{cases} (\mathbf{x}_j^k - \mathbf{x}_i^k) \frac{q_i^k q_j^k}{r_{ij}^k}, & \text{если } f(\mathbf{x}_j^k) < f(\mathbf{x}_i^k); \\ (\mathbf{x}_i^k - \mathbf{x}_j^k) \frac{q_i^k q_j^k}{r_{ij}^k}, & \text{если } f(\mathbf{x}_j^k) \geq f(\mathbf{x}_i^k), \end{cases} \quad (2)$$

где $r_{ij}^k = \|\mathbf{x}_j^k - \mathbf{x}_i^k\|$ — евклидово расстояние между i -й и j -й частицами. Интуитивно ясно, что в результате действия таких сил притяжения/отталкивания каждая частица вынуждена менять свое положение в каждый момент времени. В *EM* правило обновления положения каждой частицы задается следующим уравнением:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \lambda \widehat{F}_i^k m_i^k, \quad (3)$$

где λ — d -мерный вектор случайных чисел из интервала $[0, 1]$; $\widehat{F}_i^k = F_i^k / \|F_i^k\|$ — нормированная электростатическая сила i -й частицы на итерации k ; вектор движения $m_i^k = [m_{i,1}^k, m_{i,2}^k, \dots, m_{i,d}^k]$ зависит от $\widehat{F}_i^k = [\widehat{F}_{i,1}^k, \widehat{F}_{i,2}^k, \dots, \widehat{F}_{i,d}^k]$ согласно формуле

$$m_{i,n}^k = \begin{cases} ub_n - x_{i,n}^k, & \text{если } \widehat{F}_{i,n}^k > 0; \\ x_{i,n}^k - lb_n, & \text{если } \widehat{F}_{i,n}^k \leq 0, \end{cases} \quad (4)$$

где $x_{i,n}^k$ — n -я координата частицы \mathbf{x}_i^k ; lb_n и ub_n — нижняя и верхняя границы фитнес-функции n -й координаты соответственно.

Алгоритм гравитационного взаимодействия (*GSA*), предложенный Э. Рашеди и др. в 2009 г. [2], инспирирован законами гравитации.

В *GSA* поисковыми агентами является система из N взаимодействующих небесных тел. На

очередной итерации t алгоритма каждому i -му телу присваивается определенная масса M_i :

$$M_i^t = \frac{m_i^t}{\sum_{j=1}^N m_j^t}. \quad (5)$$

Здесь m_i^t — нормализованное значение фитнес-функции, соответствующей массе i -го тела:

$$m_i^t = \frac{f(x_i^t) - f_{worst}^t}{f_{best}^t - f_{worst}^t}, \quad (6)$$

где f_{best}^t и f_{worst}^t — текущие лучшее и худшее значения фитнес-функции в момент времени t . Согласно закону тяготения тела находятся в постоянном взаимодействии друг с другом в результате гравитационных сил. В *GSA* общая гравитационная сила, испытываемая i -м телом, задается следующим уравнением:

$$F_i^t = \sum_{j \neq i}^N \left(G(t) \frac{M_i^t M_j^t}{r_{ij}^t + \varepsilon} (\mathbf{x}_j^t - \mathbf{x}_i^t) \mathbf{rand} \right), \quad (7)$$

где $r_{ij}^t = x_j^t - x_i^t$ — евклидово расстояние между i -м и j -м телами; ε — малое значение для избежания сингулярностей; \mathbf{rand} — d -мерный вектор случайных чисел из интервала $[0, 1]$; $G(t)$ — гравитационная постоянная, значение которой зависит от текущего времени t следующим образом:

$$G(t) = G_0 e^{\left(-\frac{\alpha t}{T}\right)}, \quad (8)$$

где G_0 — начальное значение гравитационной постоянной; α — постоянный параметр; T — общее число итераций поиска оптимума алгоритмом *GSA*.

В результате гравитационного взаимодействия тела двигаются, поэтому в *GSA* применяется следующее правило обновления их положения:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \quad (9)$$

где \mathbf{v}_i^{t+1} — скорость i -го тела в момент времени $(t + 1)$, равная

$$\mathbf{v}_i^{t+1} = \mathbf{rand} \cdot \mathbf{v}_i^t + \mathbf{a}_i^t, \quad (10)$$

где $\mathbf{a}_i^t = \frac{1}{M_i^t} F_i^t$ — ускорение i -го в момент времени t .

Алгоритм моделирования отжига (*SA*), предложенный С. Киркпатриком и др. в 1982 г. [3], инспирирован физическим процессом отжига кристаллических твердых тел путем термической обработки. Твердое тело сначала нагревают до определенной температуры, а затем

ему дают медленно остыть. Этот процесс позволяет микроструктуре тела достичь лучшей конфигурации кристаллической решетки. Алгоритм начинается с определения начального решения (состояния) s и графика охлаждения $T = \{t_0, t_1, \dots, t_n\}$, где каждый элемент t_i представляет переход температуры в процессе отжига, причем $t_i > 0$ и $\lim_{i \rightarrow +\infty} t_i = 0$. Температура t_i в плане охлаждения рассчитывается для конечного числа итераций алгоритма SA. Поэтому в SA устанавливается график повторения температур $M = \{M_0, M_1, \dots, M_n\}$, где элементы M_i определяют число итераций алгоритма, при котором будет установлена заданная температура t_i . В ходе итерационного процесса поиска оптимума генерируется соседнее решение s' либо вокруг текущего наилучшего решения s^k , либо случайным образом, либо следуя определенному критерию. После этого оба решения сравниваются, и в зависимости от результата сравнения с заданной вероятностью принимается решение s' в качестве текущего наилучшего решения:

$$p^k = \begin{cases} \exp\left(-\frac{(f(s') - f(s^k))}{t^k}\right), \\ \text{если } f(s') - f(s^k) > 0; \\ 1, \text{ если } f(s') - f(s^k) \leq 0, \end{cases} \quad (11)$$

где $t^k \in T$ — температура отжига на k -й итерации.

Алгоритм синусоидальных функций (SCA), предложенный С. Мирджалили в 2016 г. [4], базируется на свойствах синусоидальных функций. В SCA каждый из N поисковых агентов характеризуется позицией $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$ в заданном d -мерном пространстве решений. Для обновления позиций агентов на очередной итерации применяется один из двух различных операторов движения, каждый из которых моделируется синусоидальной функцией (синус или косинус):

$$\mathbf{x}_i^{k+1} = \begin{cases} \mathbf{x}_i^k + r_1 \sin(r_2) |r_3 \mathbf{p}_i^t - \mathbf{x}_i^k|, \text{ если } r_4 < 0,5; \\ \mathbf{x}_i^k + r_1 \cos(r_2) |r_3 \mathbf{p}_i^t - \mathbf{x}_i^k|, \text{ если } r_4 \geq 0,5, \end{cases} \quad (12)$$

где $r_1 = a - K(a/T)$; a — константа; K — максимальное число итераций; r_2, r_3, r_4 — случайные числа из равномерно распределенных интервалов $[0, 2\pi]$, $[0, 2]$ и $[0, 1]$, соответственно; \mathbf{p}_i^t — позиция наилучшего решения.

Алгоритм перехода вещества из одного теплового состояния в другое (SMS), предложенный Э. Куэвасом и др. в 2013 г. [5], основан на физических принципах теплового движения молекул вещества при его переходе из одного состояния в другое.

В SMS поисковыми агентами являются отдельные молекулы вещества, которое может находиться в газообразном, жидком и твердом состояниях. В газообразном состоянии молекулы постоянно двигаются и сталкиваются с другими молекулами; в жидком состоянии молекулярное движение значительно уменьшается в результате снижения доступной тепловой энергии; в твердом состоянии сила связи между молекулами становится настолько сильной, что их движение почти полностью прекращается. В SMS смена фаз состояний происходит последовательно за конечное число итераций. При этом каждая молекула движется с определенной "интенсивностью", которая зависит от текущего переходного состояния. В общем случае на некоторой итерации молекула обновляет свое положение по следующему правилу:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^k \cdot \mathbf{rand} \cdot (\mathbf{ub} - \mathbf{lb})\rho, \quad (13)$$

где $\mathbf{lb} = [lb_1, lb_1, \dots, lb_d]$ и $\mathbf{ub} = [ub_1, ub_1, \dots, ub_d]$ — нижняя и верхняя границы вектора целевого пространства решений; \mathbf{rand} — случайный вектор из интервала $[0, 1]$; $\rho \in [0, 1]$ — коэффициент, значение которого зависит от текущей итерации алгоритма SMS; \mathbf{v}_i^k — скорость i -й молекулы на k -й итерации, которая задается следующим выражением:

$$\mathbf{v}_i^k = v_{init} \mathbf{d}_i^k. \quad (14)$$

Здесь v_{init} — начальная скорость молекулы:

$$v_{init} = \beta \frac{\sum_{n=1}^d (ub_n - lb_n)}{d}, \quad (15)$$

где lb_n и ub_n — нижняя и верхняя границы целевой функции в n -м измерении соответственно; $\beta \in [0, 1]$ — коэффициент, который, как и ρ в уравнении (13), зависит от текущей итерации алгоритма SMS. Величина \mathbf{d}_i^k определяет вектор направления движения молекулы и задается уравнением

$$\mathbf{d}_i^k = \mathbf{d}_i^{k-1} \cdot 0,5 \left(1 - \frac{k}{K}\right) + \mathbf{a}_i^k. \quad (16)$$

Здесь K — максимальное число итераций SMS; \mathbf{a}_i^k — единичный вектор, который определяется следующим выражением:

$$\mathbf{a}_i^k = \frac{(\mathbf{x}_{best}^k - \mathbf{x}_i^k)}{\|\mathbf{x}_{best}^k - \mathbf{x}_i^k\|}, \quad (17)$$

где \mathbf{x}_{best}^k — текущее глобальное лучшее решение.

Особенностью SMS является использование механизма столкновения молекул с последующим изменением направления их движения. Столкновение между парой молекул происходит, если евклидово расстояние между ними $\|\mathbf{x}_i - \mathbf{x}_j\|$ окажется меньше заданного порога. После столкновения молекулы обмениваются своими направления движения следующим образом:

$$\text{если } (\|\mathbf{x}_i^k - \mathbf{x}_j^k\| < r) \rightarrow \begin{cases} d_i^k = d_j^k; \\ d_j^k = d_i^k, \end{cases} \quad (18)$$

где порог расстояния r (радиус столкновения) задается выражением

$$r = \alpha \frac{\sum_{n=1}^d (ub_n - lb_n)}{d}, \quad (19)$$

где $\alpha \in [0, 1]$ — коэффициент, аналогичный β и ρ в соотношениях (13) и (15), зависящий от текущей итерации алгоритма SMS.

2. Биоэвристики, основанные на когнитивных процессах и деятельности человека

Биоэвристики, основанные на когнитивных процессах и деятельности человека, включают алгоритмы фейерверка (FWA), меметики (MA), биогеографии (BBO), поиска гармонии (HS), неокOLONиализма (ICA) и некоторые другие.

Алгоритм фейерверков (FWA), предложенный Ю. Тан и Ю. Чжу в 2010 г. [6], инспирирован процессами, наблюдаемыми при взрывах, создаваемых фейерверками.

В FWA поисковыми агентами являются искры, которые образуются после взрыва фейерверков. Хорошо изготовленные фейерверки генерируют множество централизованных искр. Число генерируемых искр s_i^k и амплитуда взрыва A_i^k для i -го фейерверка определяют его фитнес-функцию в месте взрыва \mathbf{x}_i согласно следующим выражениям:

$$s_i^k = m \frac{f_{worst}^k - f(\mathbf{x}_i^k) + \xi}{\sum_{i=1}^N f_{worst}^k - f(\mathbf{x}_i^k) + \xi}; \quad (20)$$

$$A_i^k = \hat{A} \frac{f(\mathbf{x}_i^k) - f_{best}^k + \xi}{\sum_{i=1}^N f(\mathbf{x}_i^k) - f_{best}^k + \xi}, \quad (21)$$

где m — максимальное число искр фейерверка; \hat{A} — амплитуда взрыва фейерверка; f_{best}^k и f_{worst}^k — текущие наилучшее и наихудшее значения фитнес-функции среди N фейерверков соответственно; ξ — небольшое значение, ис-

пользуемое для предотвращения сингулярностей при вычислении s_i^k или A_i^k . Чтобы избежать подавляющего эффекта "великолепного" фейерверка, границы для s_i^k определяются относительно набора постоянных параметров a и b следующим образом:

$$\hat{s}_i^k = \begin{cases} \text{round}(am), \text{ если } s_i < am; \\ \text{round}(bm), \text{ если } s_i > bm, a < b < 1; \\ \text{round}(s_i^k), \text{ иначе.} \end{cases} \quad (22)$$

Для i -го фейерверка каждая из генерируемых искр \hat{s}_i^k случайным образом распределяется по локальной области, определяемой амплитудой взрыва A_i^k . В FWA это достигается путем случайного выбора числа направлений для каждой отдельной искры, а затем вычисляется значение смещения в пределах амплитуды взрыва A_i^k .

Алгоритм меметики (MA) основан на предложенной Р. Докинзом концепции эволюции культуры [7], построенной по аналогии с генетикой и биологической теорией эволюции Дарвина и берущей начало из эволюционной модели передачи культурной информации на основе концепции мемов. Мем — единица культурной эволюции. В MA мемы, подобно генам, рассматриваются в качестве репликаторов информации, копируемой селективно. Иными словами, мемы копируются путем обучения и борются за выживание в нашей памяти и за шанс быть вновь реплицированными. Метрика мема характеризуется через его свойства: распространение, живучесть, энтропия и степень влияния [8]. Поисковыми агентами в MA является популяция $M = \{M_1, \dots, M_N\}$ из N мемов, описывающих ситуации и решения, которые признаны наиболее разумными, а также общие нормы, алгоритмы, правила, регламентирующие человеческое поведение. Для каждого мема вычисляется фитнес-функция и проводится кооперация путем обмена информацией между мемами (аналогично кроссинговеру в генетических алгоритмах). Лучший мем M_i выбирается из множества мемов на основе h запусков локального поиска. Синтаксическая модель алгоритма меметики включает следующие элементы:

$$MA = \{P_0, t, N, \delta_0, \lambda, R, F, \varphi, L\}, \quad (23)$$

где $P_0 = (P_0^1, P_0^2, \dots, P_0^N)$ — исходная популяция мемов из d -мерного дискретного пространства; t — шаг эволюции ($t = [0, 1, \dots, t_{\max}]$); N — размер популяции мемов; δ_0 — начальный набор параметров для операторов эволюции мемов; λ — размер множества мемов-потомков,

сгенерированных на очередном шаге эволюции; R — размерность задачи; F — фитнес-функция; φ — отображение, описывающее связь между мемами и кодируемыми объектами, управление и манипулирование которыми осуществляет МА; $L = \{L_1, L_2, \dots, L_m\}$ — операторы генерации мемов, используемые алгоритмом в процессе эволюции. В рамках синтаксической модели можно описать все разнообразие алгоритмов меметики при использовании различных сочетаний операторов. Алгоритмам меметики присущ параллелизм. Они проще генетических и роевых алгоритмов.

Алгоритм биогеографии (BBO), предложенный Д. Симоном в 2008 г. [9], инспирирован биогеографией — научной дисциплиной на стыке биологии и островной географии, изучающей закономерности географического распределения флоры, фауны и микроорганизмов в биосфере. В ней делается попытка предсказать число видов, существующих на островах, чтобы объяснить видовое разнообразие островов за счет миграции и мутаций. Алгоритм BBO основан на математическом аппарате для описания динамичного процесса миграции и вымирания видов на островах. Суть состоит в том, что число видов, населяющих остров, определяется равновесием между иммиграцией и вымиранием. Модель распространения видов на острове характеризуется через $\lambda(n)$ — функцию иммиграции, $\mu(n)$ — функцию эмиграции, n — число видов данного острова. При $n = 0$ эмиграция равна нулю, но с увеличением числа видов остров переполняется, и большее число видов получает стимулы для того, чтобы покинуть этот остров и исследовать другие острова. Равенство уровней иммиграции и эмиграции достигается при некотором числе видов n_0 , которое случайно колеблется в зависимости от изменений климата острова, эпидемий и пр. Если обозначить $p_n(t)$ — вероятность того, что рассматриваемый остров содержит n видов в момент времени t , то можно оценить изменение вероятности p_n за период времени $[t, t + \Delta t]$:

$$\frac{dp_n(t)}{dt} \cong \lim_{\Delta t \rightarrow 0} \frac{p_n(t + \Delta t) - p_n(t)}{\Delta t}. \quad (24)$$

Чтобы в момент времени $(t + \Delta t)$ на острове содержалось n видов, должно выполняться одно из следующих условий: в момент времени t остров содержал n видов, а иммиграция и эмиграция за период Δt отсутствовала; в момент времени t остров содержал $(n - 1)$ видов, а за период Δt иммигрировал один вид; в момент времени t остров содержал $(n + 1)$ видов, а за

период Δt эмигрировал один вид. В пределе при $\Delta t \rightarrow 0$ вероятность более чем одной иммиграции или эмиграции очень мала. Получаем

$$p(t + \Delta t) \cong p_n(t) + \dot{p}_n(t)\Delta t. \quad (25)$$

Формула (25) является основным расчетным соотношением, используемым в биогеографическом алгоритме, а фитнес-функцией — индекс пригодности острова [10].

Алгоритм поиска гармонии (HS), предложенный З. Гаэм и др. в 2001 г. [11], основан на принципах, лежащих в основе процесса гармонической импровизации, когда музыканты комбинируют различные тона с целью найти идеальную гармонию. Алгоритм инициализируется путем случайного выбора из гармонической памяти N решений: $\mathbf{x}_c = [x_{c,1}, x_{c,2}, \dots, x_{c,d}]$, затем каждое решение оценивается, и выбираются лучшие из них. Каждый компонент $x_{c,n}$ импровизированного решения генерируется в зависимости от значения гармонии $HMCR$, как показано ниже:

$$x_{c,n} = \begin{cases} x_{r,n} + rand(-1,1) \cdot bw, & \text{если } rand(0,1) < HMCR; \\ rand(lb_n, ub_n), & \text{если } rand(0,1) \geq HMCR, \end{cases} \quad (26)$$

где $x_{r,n}$ — n -я компонента, соответствующая случайному выбранному решению \mathbf{x}_r в пределах гармонической памяти; bw — параметр полосы пропускания; $rand(a, b)$ — случайное число из интервала $[a, b]$; lb_n и ub_n — нижняя и верхняя границы фитнес-функции.

Алгоритм неокOLONИализма (ICA), предложенный Ф. Аташпаз-Гаргари и К. Лукасом в 2007 г. [12], основан на действиях, предпринимаемых отдельными странами для расширения своей власти, как правило, путем колонизации других территорий.

На этапе инициализации ICA начинается со случайной генерации популяции из N агентов (страны), каждый из которых представляет решение $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$ как совокупность социально-политических характеристик (культура, язык, экономика, религия и т. д.). Затем множество решений в зависимости от фитнес-функции классифицируются как империалистические страны N_{imp} : $(\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_{N_{imp}})$ и страны-колонии $N_{col} = N - N_{imp}$: $(\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_{N_{col}})$. Каждая из N_{col} колоний пропорционально распределяется между N_{imp} странами для формирования империй. Число колоний в империи пропорционально значениям фитнес-функций империалистической страны согласно формуле

$$power_i = \left| \frac{cost_i}{\sum_{j=1}^{N_{imp}} cost_j} \right|, \quad (27)$$

где

$$cost_i = \max_j \{f(\mathbf{i}_j)\} - f(\mathbf{i}_i). \quad (28)$$

Число колоний, закрепленных за каждой империалистической страной, определяется как

$$NC_i = round(power_i \cdot N_{col}). \quad (29)$$

Итеративный процесс *ICA* включает этапы ассимиляции, революции и конкуренции. Ассимиляция предполагает, что каждая колония, входящая в состав империи, находится под влиянием социально-политических факторов соответствующей империалистической страны. Указанное влияние представляется как движение колонии \mathbf{c}_{ij} к \mathbf{i}_j согласно формуле

$$\mathbf{c}_{ij}^* = \mathbf{c}_{ij} + rand(0, \beta)(\mathbf{i}_j - \mathbf{c}_{ij}), \quad (30)$$

где $\beta \in [1, 2]$. Затем, на этапе революции, предполагается, что некоторые случайно выбранные колонии испытывают изменения в своих социально-политических характеристиках (аналогично оператору мутации в *GA*). Колония \mathbf{c}_{ij} после некоторых изменений может превратиться в империалистическую, и наоборот. На этапе конкуренции слабая колония внутри слабой империи может передаваться другой империи с вероятностью

$$P_j = \left| \frac{NTC_j}{\sum_{i=1}^{N_{imp}} NTC_i} \right|, \quad (31)$$

где NTC_j — нормализованная общая стоимость j -й империи:

$$NTC_j = \max_i \{TC_j\} - TC_j, \quad (32)$$

$$TC_j = f(\mathbf{i}_j) + \xi \cdot mean\{f(\mathbf{c}_{ij})\},$$

$\xi \in [0, 1]$ — параметр, отражающий влияние, которое оказывают власти колонии на власти империи. В результате конкуренции более слабые империи постепенно теряют свои колонии, а более сильные империи овладевают новыми колониями. Более слабые империи со временем рушатся, пока не останется только одна самая сильная империя.

3. Сравнительные характеристики биоэвристик

Проведем сопоставительный анализ таких характеристик рассмотренных биоэвристик, как баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений, вычислительная трудоемкость, требуемый объем памяти, настройка параметров алгоритма, программная реализация.

Баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений. Диверсификация пространства поиска решений относится к способности поисковых агентов исследовать новые области пространства поиска, в то время как сходимость алгоритма делает акцент на возможностях этих агентов уточнять уже найденные "хорошие" решения. Важность поиска баланса между скоростью сходимости алгоритма и диверсификацией пространства поиска решений является фундаментальной задачей, однако отсутствуют инструменты его измерения. По нашему мнению, оценка баланса между скоростью сходимости алгоритма и диверсификацией пространства поиска решений, в основном, зависит от используемого в биоэвристике механизма популяционного отбора, аттрактивности операторов поиска, настройки параметров и от числа итераций алгоритма.

Популяционные биоэвристики используют различные механизмы отбора, позволяют проводить селекцию лучших решений на очередной итерации, обновляя популяцию. Операторы селекции не изменяют и не генерируют новых решений в популяции, однако они играют важную роль в балансе между элитарностью и разнообразием популяции. В случае жадного механизма отбора это может привести к преждевременной сходимости в точке локального оптимума. Анализ показывает, что жадный механизм (быстрое принятие локально оптимальных решений на каждом этапе) характерен для биоэвристик *GA* и *FWA*, в отличие от таких биоэвристик, как *ES*, *PSO*, *FA*, *CSA*, *FPA*, *GWO*, *WOA*, *GSA*, *EM*, *SCA*, *SMS*, *BCA*, *BBO*, которые используют механизм, отличный от жадного отбора. Несколькими иной тип элитарности (индивидуальный жадный отбор) наблюдается в алгоритмах *DE* и *HS*. В них новое решение выбирается для дальнейшего поиска, только если оно улучшает предыдущее решение.

Аттрактивность операторов, используемых в биоэвристике, означает дрейф имеющихся решений в направлении лучших решений, которые рассматриваются в качестве аттракторов. Например, в *PSO* "движение" всех частиц направлено к наилучшему глобальному решению

на текущей итерации процесса поиска, однако с учетом хранящейся информации в каждой частице о личном лучшем решении. В *FA* притяжение между светлячками обратно пропорционально расстоянию, которое их разделяет, и чем больше расстояние, тем меньше притяжение. Аналогично, в *GSA* гравитационная сила, действующая между небесными телами, также обратно пропорциональна расстоянию между ними. Эти механизмы эффективны для поддержания разнообразия популяции, однако их моделирование требует постоянного вычисления расстояния между несколькими парами решений, увеличивая вычислительную сложность алгоритмов. Такого рода механизмы замедляют скорость сходимости алгоритма, увеличивая время их выполнения. Некоторые алгоритмы не рассматривают какой-либо тип притяжения как часть своей стратегии поиска; вместо этого они генерируют решения с помощью случайных ходов, например *HS*. В некоторых модификациях *GA* решения генерируются путем "скрещивания" со случайно выбранным решением (кроссинговер), либо мутацией в существующих решениях. Наблюдается большое разнообразие вариантов оператора кроссинговера и мутации. Затруднительно однозначно классифицировать операторы кроссинговера и мутации как операторы, направленные либо на увеличение скорости сходимости алгоритма, либо на диверсификацию пространства поиска решений.

Настройка параметров биоэвристики также позволяет изменить баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений. Однако универсального ответа на вопрос об оптимальном выборе параметров биоэвристики для решения конкретной задачи оптимизации нет. Чаще всего это делается вручную, руководствуясь опытом и некоторыми эмпирическими правилами [13]. К параметрам биоэвристики относятся параметры их остановки: сформировано заданное число поколений; исчерпано время, отведенное на работу алгоритма; популяция достигла заданного качества; достигнут некоторый уровень сходимости (особи в популяции стали настолько подобными, что дальнейшее их улучшение происходит чрезвычайно медленно) и т. д. Эти параметры также влияют на баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений.

Вычислительная трудоемкость. Оценка вычислительной трудоемкости зависит от сортировки популяции и трудоемкости вычисления фитнес-функции. В зависимости от стратегии поиска в некоторых биоэвристиках на каждой итерации требуется сортировка популяции ре-

шений в зависимости от значений их фитнес-функции, чтобы оставить в популяции лучшие решения (например, в *GA*, *GWO*, *FWA*). Сортировка популяции может быть реализована с помощью процедур с различной вычислительной сложностью. Например, в *MATLAB* по умолчанию используется сортировка с вычислительной сложностью $O(N \log(N))$, где N — число индивидов в популяции. В других биоэвристиках требуется проводить некоторые вычисления, например, измерять евклидово расстояние между индивидами в популяции в некоторой окрестности пространства поиска (*GSA*, *SMS*, *SSO*, *EM*, *FWA*, *BCA*). Евклидовы расстояния могут вычисляться для каждой отдельной пары решений в популяции (наихудший сценарий), что потребует, по крайней мере, $O(N^2)$ затрат на операцию вычисления квадратного корня. Достаточно трудоемкой является процедура вычисления фитнес-функций в *ABC*, *FA*, *FWA*, *BCA*. Обычно в биоэвристиках вычисление фитнес-функции выполняется фиксированное число раз, однако в *ABC*, *FA*, *FWA*, *BCA* это правило нарушается.

Требуемый объем памяти. Оценка пространственной сложности биоэвристики зависит от размера популяции, а также от объема памяти, требуемой для хранения данных. Все популяционные биоэвристики требуют минимальной памяти размером $O(N(D + 1))$, где D — размерность каждого индивида, для хранения индивидов популяции и значений их фитнес-функций. Если запоминать лучшее и худшее решения, а также значения их фитнес-функций, то требуемый минимальный объем памяти будет равен

$$O(N(D + 1) + a(D + 1)),$$

где $a \in \{1, 2\}$. Кроме того, некоторые алгоритмы требуют дополнительной памяти для хранения данных в зависимости от используемой в них стратегии поиска. Например, в алгоритмах *SMS*, *KH* и *GSA* необходимо вычислять и хранить в памяти евклидовы расстояния между индивидами популяции. Алгоритм *ICA* требует вычисления расстояния между колониями и империей и хранения этих результатов. Для многомерных задач и больших данных эти проблемы могут ограничить их применение на компьютерах с ограниченными ресурсами.

Настройка параметров алгоритма. Оценка настройки параметров биоэвристики зависит от числа параметров, от их влияния на производительность алгоритма. Простая настройка параметров отличает следующие биоэвристики: *GA*, *DE*, *PSO*, *BA*, *CSA*, *CS*, *FPA*, *GSA*, *HS*, в отличие

от *ABC*, *GWO*, *EM*. Нет универсального ответа на вопрос об оптимальном выборе параметров биоэвристики для решения конкретной задачи оптимизации. Чаще всего это делается вручную, на основе опыта и некоторых эмпирических правил. Например, в отличие от алгоритмов *GA* или *PSO*, которые содержат небольшое число параметров, такие алгоритмы, как *ES*, *FA*, *SSO*, *KHA*, *SMS*, *FWA*, *BCA* или *SMS*, требуют от пользователя установки нескольких параметров, необходимых для управления работой алгоритма. При этом по мере увеличения числа устанавливаемых параметров понимание того, как эти значения влияют на производительность алгоритма, становится более сложным, что делает алгоритм более трудным для анализа в целом. Правда, существуют алгоритмы, которые не имеют никаких параметров настройки: алгоритм группы китов *WOA* и алгоритм синусоидальных функций *SCA*. У них интегрированы механизмы для автоматического баланса между скоростью сходимости алгоритма и диверсификацией пространства поиска, пользователю нет необходимости устанавливать какие-либо параметры для его работы.

Программная реализация. Сравнение трудоемкости программной реализации биоэвристик показывает, что некоторые алгоритмы относительно просто программируются: *GA*, *DE*, *ES*, *PSO*, *BA*, *CSA*, *FPA*, *GWO*, *WOA*, *SCA*. Алгоритм *SCA* можно считать одним из самых простых для кодирования, поскольку при поиске наилучшего решения не требуется сортировка популяции. Число строк кода биоэвристики возрастает при усложнении механизмов и процедур, интегрированных в эти алгоритмы: *SSO*, *SMS*, *FWA*. Алгоритм *HS* просто программируется на этапе инициализации решений-кандидатов, однако использование в *HS* жадного механизма отбора требует выполнения сортировки популяции, что увеличивает сложность его кодирования и, как следствие, его вычислительную сложность.

4. Приложения биоэвристик

В последние годы биоэвристики стали активно применяться для решения широкого спектра задач в таких областях, как инженерное проектирование, цифровая обработка изображений и компьютерное зрение, сети и коммуникации, энергетика и энергоменеджмент, анализ и обработка данных, машинное обучение, робототехника, медицинская диагностика и многие другие. Рассмотрим несколько примеров применения биоэвристик в этих областях.

В области инженерного проектирования существуют разнообразные примеры успешного применения биоэвристик. В работе [14] представлен пример применения *GA*, *DE* и *PSO* при проектировании антенн. В статье [15] авторы предложили оптимизировать конструкции авиационных панелей на основе алгоритма *HS*, в [16] представлен алгоритм *FA* для задачи выбора параметров в процессе механической обработки деталей, в [17] приводятся примеры проектирования пружинных конструкций, сварных балок, труб высокого давления, зубчатых передач и др.

Примером применения биоэвристик *ACO*, *PSO*, *ABC*, *SSO*, *FPA*, *WOA*, *FA*, *GWO* в области цифровой обработки изображений и компьютерного зрения являются задачи сегментации изображений, обнаружения областей интереса на цифровых изображениях, цветовой экстракции [18–21].

В работах [22, 23] алгоритмы *GA*, *ABC* и *PSO* были успешно применены для решения задачи оптимального распределения сети сенсорных датчиков для сбора физических данных. Другими примерами являются решение задач обнаружения сообществ в сложных сетевых структурах с помощью биоэвристик *PSO* и *GA* [24, 25], применение биоэвристики *ABC* для обнаружения вредоносных *URL*-адресов [26], алгоритма *CS* для выполнения криптоанализа в области информационной безопасности [27], алгоритма *PSO* для обнаружения спама [28].

В работе [29] авторы предложили оптимальную стратегию энергоменеджмента для гибридных систем хранения энергии на основе *PSO* и симплекс-метода. В статье [30] приводятся результаты алгоритма *SSO* для управления домашней энергоустановкой. В [31] представлен подход, основанный на алгоритме *GWO*, направленный на решение задачи регулирования нагрузки в сети энергосистем путем настройки параметров ПИД регулятора. В [32] алгоритм криля *KHA* был успешно реализован для решения задачи оптимального управления мощностью в энергосистеме. В [33] алгоритм интеллектуального движения капель воды и биогеографический алгоритм *BBO* был применен для диспетчеризации нагрузки в электросети, в [34] с помощью алгоритма *PSO* решалась задача реконфигурации фотоэлектрических панелей. В [35] алгоритм *WOA* применялся для проблемы оптимального размещения конденсаторов в радиальных распределительных сетях, в [36] с помощью алгоритма *FA* решалась задача оптимального размещения ветрогенераторов.

Интересным приложением в области интеллектуального анализа данных являются алго-

ритмы *WOA*, *GWO*, *SA*, *PSO* и *GA*, представленные в работах [37, 38] для задач выбора информативных признаков для классификации объектов и ранжирования многомерных данных. Другие приложения биоэвристик *GSA*, *ABC*, *PSO* и *FA* можно найти в статьях [39–41] для задачи кластеризации данных, обучения сверточных нейросетей, для настройки параметров в *SVM*-методе машинного обучения.

Реализация биоэвристик в области робототехники включает в себя такие задачи, как планирование траектории и оптимизация траектории роботов. Наиболее популярным применением в этом плане является автономная навигация беспилотных летательных аппаратов. В работе [42] представлен алгоритм *GSA*, а в [43] — алгоритм *SA* для разработки стратегии планирования полета беспилотника. В статье [44] описана разработка контроллеров для роботизированных платформ, в которых был применен алгоритм *GP* для управления роботом.

Применение биоэвристик в области медицинской диагностики, в основном, связано с обработкой медицинских изображений. Например, в работе [45] был предложен подход к сегментации МРТ-изображений, основанный на алгоритмах *ACO* и *PSO*. Другие применения природных методов в медицинской диагностике включают разработку инструментария, направленного на прогнозирование заболеваний путем анализа больших наборов данных с использованием алгоритма *BA* [46]. В статье [47] алгоритм *ABC* использовался для помощи в решении задачи классификации онкологии с использованием наборов данных микрочипов.

5. Результаты экспериментов

Чтобы продемонстрировать эффективность, а также вычислительные характеристики некоторых из представленных выше биоэвристик,

Таблица 1

Набор тестовых графов *Peco*

№ теста	Размер решетки	Число элементов	Число соединений
1	10×10	100	184
2	95×95	9025	17 864
3	100×100	10 000	19 804
4	190×190	36 100	71 824
5	200×200	40 000	79 604

были проведены вычислительные эксперименты на наборе тестовых схем *Peco*, представленных в табл. 1. В качестве тестовой задачи использовалась задача размещения графа в разные решетки по критерию минимальной суммарной длины соединений, измеряемых в микрометрах [51, 52].

Для проведения экспериментальных исследований были выбраны и программно реализованы следующие алгоритмы: *KraftWerk*, *Capo 8.6*, эволюционных стратегий *ES*, генетический *GA*, меметический *MA*, биогеографический *BBO*, моделирования отжига *SA*, пчелиный *ABC* и светлячковый *FA*.

При построении комплекса программ использовались пакеты *Visual C++*, *Borland C++*, *Builder*. Отладка и тестирование разработанных алгоритмов выполнялось на компьютере типа *IBM PC* с процессором *ryzen 5 3600x* с ОЗУ 16 Гбайт. Селекция использовалась на основе колеса рулетки, начальная популяция 50 решений и 100 итераций алгоритмов.

Приведем зависимость качества работы данных алгоритмов от тестовых графов. Результаты вычислительного эксперимента по оптимизации значений фитнес-функции по критерию *HPWL* (*HPWL* — суммарная длина соединений, полученная при размещении в разные решетки измеряемая в микрометрах) представлены в табл. 2.

Таблица 2

Результаты оптимизации размещения тестовых графов *Peco* по критерию *HPWL* алгоритмами *KraftWerk*, *Capo 8.6*, *ES*, *GA*, *MA*, *BBO*, *SA*, *ABC*, *FA*

№ теста	Оптимум	<i>KraftWerk</i>	<i>Capo8.6</i>	<i>ES</i>	<i>GA</i>	<i>MA</i>	<i>BBO</i>	<i>SA</i>	<i>ABC</i>	<i>FA</i>
	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>	<i>HPWL</i>
1	184	202	184	205	190	198	200	186	189	192
2	17884	18302	22764	22654	18246	18298	18276	18257	18251	18242
3	19804	20519	21314	21645	20645	20697	20713	21802	20594	20635
4	71864	75384	77814	77753	75356	75414	75485	77413	75405	75386
5	79604	82335	88041	88356	82184	82236	82987	85314	82229	82274

Сравнение результатов показывает, что алгоритм *ES* быстрее других конкурирующих алгоритмов выполняет размещение, но значительно уступает им по качеству размещения графа в решетке. Остальные биоэвристики по качеству и времени решения сопоставимы с известными алгоритмами *KraftWerk* и *Capo8.6*.

Заключение

Лучшим способом подтверждения эффективности биоэвристики является доказательство сходимости и оценка вычислительной сложности. Как правило, это возможно только в случае упрощенной постановки задачи. Альтернативой является проверка алгоритмов на тестовых задачах (бенчмарках) в проблемных областях. В настоящее время не существует согласованного каталога таких задач для оценки биоэвристик, хотя для отдельных задач они уже сложились и широко используются. Например, интересный сравнительный анализ алгоритмов *PSO*, *ACO*, *CS*, *FPA*, *GSA* можно найти в работах [48, 49] для решения задач коммивояжера и маршрутизации транспортного средства. Исследователи предлагают интересные идеи, направленные на улучшение производительности биоэвристик, например, гибридизацию точных методов с биоэвристиками в целях повышения их эффективности и сходимости [50]. Однако результаты измерения свойств полученных гибридов, оценки их производительности, валидация таких показателей, как ошибка, среднее, медианное и стандартное отклонение, отсутствуют.

В целом поиск баланса между скоростью сходимости биоэвристик и диверсификацией пространства поиска решений является актуальной исследовательской проблемой, имеющей важное значение для обеспечения точности и производительности биоэвристик.

Список литературы

1. **Birbil S., Fang S.** An electromagnetism-like mechanism for global optimization // *Jour. Glob. Optim.* 2003. Vol. 25, N.3. P. 263–282.
2. **Rashedi E., Nezamabadi H., Saryzadi S.** GSA: a gravitational search algorithm // *Inf. Sci.* 2009. Vol. 179, N. 13. P. 2232–2248.
3. **Kirkpatrick S., Gelatt C., Vecch M.** Optimization by simulated annealing // *Science.* 2007. Vol. 220. P. 671–680.
4. **Mirjalili S.** SCA: A Sine Cosine Algorithm for solving optimization problems // *Knowl. based Syst.* 2016. Vol. 96. P. 120–133.
5. **Cuevas E., Echavarría A., Ramírez-Ortegón M.** An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation // *Appl. Intell.* 2013. Vol. 40(2). P. 256–272.

6. **Tan Y., Zhu Y.** Fireworks algorithm for optimization // *Proc. First Int. Conf. (ICSI).* 2010. Part I. P. 355–364.
7. **Доккинз Р.** Эгоистичный ген. М.: АСТ CORPUS, 2013. 472 с.
8. **Rodzin S., Rodzina O.** Effectiveness evaluation of metrics and biogeography algorithms using benchmark and trans computational tasks // *Proc. of the First Int. Scien. Conf. ИИТ'16.* 2016. Vol. 1. P. 463–475.
9. **Simon D.** Biogeography-based optimization // *IEEE Trans. on Evolutionary Computation.* 2008. Vol. 12. № 6. P. 702–713.
10. **Rodzin S., Rodzina O.** New computational models for big data and optimization // *Proc. of the 9th Int. Conf. on AICT'15.* 2015. P. 3–7.
11. **Geem Z., Kim J., Loganathan G.** A new heuristic optimization algorithm: harmony search // *Simulation.* 2001. Vol. 76(2). P. 60–68.
12. **Atashpaz-Gargari E., Lucas C.** Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition // *Proc. IEEE Congress on Evolutionary Computation (CEC07).* 2007. P. 4661–4667.
13. **Yang X.** Swarm-based metaheuristic algorithms and no-free-lunch theorems // *Intech Open.* 2018. N. 2. P. 64–73.
14. **Goudos S.** Antenna design using binary differential evolution // *IEEE Antennas Propag. Mag.* 2017. Vol. 59(1). P. 74–93.
15. **Keshtegar B.** et. al. Optimum design of aircraft panels based on adaptive dynamic harmony search // *Thin-Walled Struct.* 2017. Vol. 118. P. 37–45.
16. **Shukla R., Singh D.** Selection of parameters for advanced machining processes using firefly algorithm // *Eng. Sci. Technol.* 2016. Vol. 20(1). P. 1–10.
17. **Camarena O.** et. al. Ls-II: An improved locust search algorithm for solving constrained optimization problems // *Mathemat. Problems in Engin.* 2018. DOI: 10.1155/2018/4148975.
18. **Mesejo P.** et. al. A survey on image segmentation using metaheuristic-based deformable models: critical analysis // *Appl. Soft Comput.* 2016. Vol. 44. P. 1–29.
19. **Khairuzzaman A., Chaudhury S.** Multilevel thresholding using grey wolf optimizer for image segmentation // *Expert Syst. Appl.* 2017. Vol. 86. P. 64–76.
20. **El-Aziz M., Ewees A., Hassanien A.** Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation // *Expert Syst. Appl.* 2017. Vol. 83. P. 242–256.
21. **He L., Huang S.** Modified firefly algorithm based multilevel thresholding for color image segmentation // *Neurocomputing.* 2017. Vol. 240. P. 152–174.
22. **Deif D., Member S.** An ant colony optimization approach for the deployment of reliable wireless sensor networks // *IEEE Access.* 2017. N. 5. P. 744–756.
23. **Alia O.** Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm // *IEEE Sens. J.* 2017. Vol. 17(3). P. 882–896.
24. **Rahimi S., Abdollahpouri A., Moradi P.** A multi-objective particle swarm optimization algorithm for community detection in complex networks // *Swarm Evol. Comput.* 2018. Vol. 39. P. 297–309.
25. **Guerrero M.** et. al. Adaptive community detection in complex networks using genetic algorithms // *Neurocomputing.* 2017. Vol. 266. P. 101–113.
26. **Bhardwaj T., Sharma T., Pandit M.** Social engineering prevention by detecting malicious URLs using artificial bee colony algorithm // *Proc. 3rd Int. Conf. on Soft Comp. for Problem Solving.* 2014. P. 355–363.
27. **Din M.** et. al. Applying Cuckoo search for analysis of LFSR based cryptosystem // *Perspect. Sci.* 2016. N. 8. P. 435–439.
28. **Johny D.** Negative selection algorithm: a survey // *Int. J. Sci. Eng. Technol. Res.* 2017. N. 6(4). P. 711–715.
29. **Mesbahi T.** et. al. Optimal energy management for a Li-ion battery/supercapacitor hybrid energy storage system based on particle swarm optimization incorporating Nelder-Mead simplex approach // *IEEE Trans. Intell. Veh.* 2017. N. 2(2). P. 1–11.

30. **You I., Yim K.** A social spider optimization-based home energy management system // Proc. Int. Conf. on Network-Based Information Systems. 2017. P. 771–778.
31. **Guha D., Roy P.** Load frequency control of interconnected power system using grey wolf optimization // Swarm Evol. Comput. 2016. Vol. 27. P. 97–115.
32. **Prasad D., Mukherjee A., Mukherjee V.** Application of chaotic krill herd algorithm for optimal power flow with direct current link placement problem // Chaos, Solitons Fractals. 2017. Vol. 103. P. 90–100.
33. **Rodzin S.** Smart Dispatching and Metaheuristic Swarm Flow Algorithm // J. of Computer and Systems Sc. Int. 2014. Vol. 53. N. 1. P. 109–115.
34. **Babu T. et. al.** Particle swarm optimization based solar PV array reconfiguration of the maximum power extraction under partial shading conditions // IEEE Trans. Sustain. Energy. 2017. Vol. 30. P. 129–138.
35. **Prakash D., Lakshminarayana C.** Optimal siting of capacitors in radial distribution network using whale algorithm // Alexandria Eng. J. 2016. N. 8. P. 57–68.
36. **Massan S. et. al.** Wind turbine micro siting by using the firefly algorithm // Appl. Soft Comput. J. 2015. Vol. 27. P. 450–456.
37. **Mafarja M., Mirjalili S.** Hybrid whale optimization algorithm with simulated annealing for feature selection // Neurocomputing. 2016. Vol. 260. P. 302–312.
38. **Moayedikia A. et. al.** Feature selection for high dimensional imbalanced class data using harmony search // Eng. Appl. Artif. Intell. 2017. Vol. 57. P. 38–49.
39. **Alswaitti M., Albughdadi M.** Density-based particle swarm optimization algorithm for data clustering // Expert Syst. Appl. 2018. Vol. 91. P. 170–186.
40. **Abualigah L. et. al.** A novel hybridization strategy for krill herd algorithm applied to clustering techniques // Appl. Soft Comput. 2017. Vol. 60. P. 423–435.
41. **Abualigah L., Khader A.** Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering // Supercomputing. 2017. Vol. 73(11). P. 4773–4795.
42. **Li P., Duan H.** Path planning of unmanned aerial vehicle based on improved gravitational search algorithm // Chin Technol. Sci. 2012. Vol. 55(10). P. 2712–2719.
43. **Oz I., Topcuoglu H.** A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles // Simulation. 2013. Vol. 89(8). P. 903–920.
44. **Silva P. et. al.** Automatic generation of biped locomotion controllers using genetic programming // Rob. Auton. Syst. 2014. Vol. 62(10). P. 1531–1548.
45. **Родзин С. И., Скобцов Ю. А., Эль-Хатиб С. А.** Биоэвристики: теория, алгоритмы и приложения. Чебоксары: ИД "Среда", 2019. 224 с.
46. **Wang K. et. al.** An improved electromagnetism-like mechanism algorithm and its application to the prediction of diabetes // J. Biomed. Inform. 2015. Vol. 54. P. 220–229.
47. **Alshamlan H., Badr G., Alohal Y.** MRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling // Biomed Res. Int. 2015. Vol. 23. P. 201–215.
48. **Saji Y., Riffi M.** A novel discrete algorithm for solving the travelling salesman problem // Neural Comput. Appl. 2016. Vol. 27(7). P. 1853–1866.
49. **Wei L.** A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints // Eur. J. Oper. Res. 2017. N. 5. P. 1–17.
50. **Jourdan L., Basseur M., Talbi E.** Hybridizing exact methods and metaheuristics: a taxonomy // Eur. J. Oper. Res. 2009. Vol. 199(3). P. 620–629.
51. **Kureichik V., Kuliev E., Kureichik V.** Monkey search algorithm for ECE components partitioning // J. of Physics: Conf. Series. 2018. Vol. 1015 (4), N. UNSP 042026. P. 1–10.
52. **Kureichik V., Kuliev E., Kureichik V.** Mechanisms of swarm intelligence and evolutionary adaptation for solving PCB design tasks // Proc. Int. Sem. on Electron Devices Design and Production (SED). 2019. P. 109–113.

V. V. Kureychik, Professor, e-mail: vkur@sfnu.ru, **S. I. Rodzin**, Professor, e-mail: srodzin@yandex.ru, Southern Federal University, Taganrog, Russian Federation

Computational Models of Bio Heuristics Based on Physical and Cognitive Processes (Review)

Computational models of bio heuristics based on physical and cognitive processes are presented. Data on such characteristics of bio heuristics (including evolutionary and swarm bio heuristics) are compared.) such as the rate of convergence, computational complexity, the required amount of memory, the configuration of the algorithm parameters, the difficulties of software implementation. The balance between the convergence rate of bio heuristics and the diversification of the search space for solutions to optimization problems is estimated. Experimental results are presented for the problem of placing Peco graphs in a lattice with the minimum total length of the graph edges.

Keywords: bio heuristics, convergence of the algorithm, exploration versus exploitation, population, fitness assignment, optimization, graph

DOI: 10.17587/it.27.563-574

References

- Birbil S., Fang S.** An electromagnetism-like mechanism for global optimization, *Jour. Glob. Optim.*, 2003, vol. 25(3), pp. 263–282.
- Rashedi E., Nezamabadi H., Saryazdi S.** GSA: a gravitational search algorithm, *Inf. Sci.*, 2009, vol. 179(13), pp. 2232–2248.
- Kirkpatrick S., Gelatt C., Vecch M.** Optimization by simulated annealing, *Science*, 2007, vol. 220, pp. 671–680.
- Mirjalili S.** SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowl. based Syst.*, 2016, vol. 96, pp. 120–133.
- Cuevas E., Echavarría A., Ramírez-Ortegón M.** An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation, *Appl. Intell.*, 2013, vol. 40(2), pp. 256–272.
- Tan Y., Zhu Y.** Fireworks algorithm for optimization, *Proc. First Int. Conf. (ICSI)*, 2010, part I, pp. 355–364.

7. **Dawkins R.** The Selfish Gene, Moscow, AST CORPUS, 2013, 472 p.
8. **Rodzin S., Rodzina O.** Effectiveness evaluation of memetics and biogeography algorithms using benchmark and trans computational tasks, *Proc. of the First Int. Scien. Conf. IITI'16*, 2016, vol. 1, pp. 463–475.
9. **Simon D.** Biogeography-based optimization, *IEEE Trans. on Evolutionary Computation*, 2008, vol. 12, no. 6, pp. 702–713.
10. **Rodzin S., Rodzina O.** New computational models for big data and optimization, *Proc. of the 9th Int. Conf. on AICT'15*, 2015, pp. 3–7.
11. **Geem Z., Kim J., Loganathan G.** A new heuristic optimization algorithm: harmony search, *Simulation*, 2001, vol. 76(2), pp. 60–68.
12. **Atashpaz-Gargari E., Lucas C.** Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *Proc. IEEE Congress on Evolutionary Computation (CEC07)*, 2007, pp. 4661–4667.
13. **Yang X.** Swarm-based metaheuristic algorithms and no-free-lunch theorems, *Intech Open*, 2018, no. 2, pp. 64–73.
14. **Goudos S.** Antenna design using binary differential evolution, *IEEE Antennas Propag. Mag.*, 2017, vol. 59(1), pp. 74–93.
15. **Keshtegar B.** et al. Optimum design of aircraft panels based on adaptive dynamic harmony search, *Thin-Walled Struct.*, 2017, vol. 118, pp. 37–45.
16. **Shukla R., Singh D.** Selection of parameters for advanced machining processes using firefly algorithm, *Eng. Sci. Technol.*, 2016, vol. 20(1), pp. 1–10.
17. **Camarena O.** et al. Ls-II: An improved locust search algorithm for solving constrained optimization problems, *Mathemat. Problems in Engin.*, 2018, DOI: 10.1155/2018/4148975.
18. **Mesejo P.** et al. A survey on image segmentation using metaheuristic-based deformable models: critical analysis, *Appl. Soft Comput.*, 2016, vol. 44, pp. 1–29.
19. **Khairuzzaman A., Chaudhury S.** Multilevel thresholding using grey wolf optimizer for image segmentation, *Expert Syst. Appl.*, 2017, vol. 86, pp. 64–76.
20. **El-Aziz M., Ewees A., Hassanien A.** Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation, *Expert Syst. Appl.*, 2017, vol. 83, pp. 242–256.
21. **He L., Huang S.** Modified firefly algorithm based multilevel thresholding for color image segmentation, *Neurocomputing*, 2017, vol. 240, pp. 152–174.
22. **Deif D., Member S.** An ant colony optimization approach for the deployment of reliable wireless sensor networks, *IEEE Access*, 2017, no. 5, pp. 744–756.
23. **Alia O.** Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm, *IEEE Sens. J.*, 2017, vol. 17(3), pp. 882–896.
24. **Rahimi S., Abdollahpouri A., Moradi P.** A multi-objective particle swarm optimization algorithm for community detection in complex networks, *Swarm Evol. Comput.*, 2018, vol. 39, pp. 297–309.
25. **Guerrero M. et al.** Adaptive community detection in complex networks using genetic algorithms, *Neurocomputing*, 2017, vol. 266, pp. 101–113.
26. **Bhardwaj T., Sharma T., Pandit M.** Social engineering prevention by detecting malicious URLs using artificial bee colony algorithm, *Proc. 3rd Int. Conf. on Soft Comp. for Problem Solving*, 2014, pp. 355–363.
27. **Din M.** et al. Applying Cuckoo search for analysis of LFSR based cryptosystem, *Perspect. Sci.*, 2016, no. 8, pp. 435–439.
28. **Johny D.** Negative selection algorithm: a survey, *Int. J. Sci. Eng. Technol. Res.*, 2017, no. 6(4), pp. 711–715.
29. **Mesbahi T.** et al. Optimal energy management for a Li-ion battery/supercapacitor hybrid energy storage system based on particle swarm optimization incorporating Nelder-Mead simplex approach, *IEEE Trans. Intell. Veh.*, 2017, no. 2(2), pp. 1–11.
30. **You I., Yim K.** A social spider optimization-based home energy management system, *Proc. Int. Conf. on Network-Based Information Systems*, 2017, pp. 771–778.
31. **Guha D., Roy P.** Load frequency control of interconnected power system using grey wolf optimization, *Swarm Evol. Comput.*, 2016, vol. 27, pp. 97–115.
32. **Prasad D., Mukherjee A., Mukherjee V.** Application of chaotic krill herd algorithm for optimal power flow with direct current link placement problem, *Chaos, Solitons Fractals*, 2017, vol. 103, pp. 90–100.
33. **Rodzin S.** Smart Dispatching and Metaheuristic Swarm Flow Algorithm, *J. of Computer and Systems Sc. Int.*, 2014, vol. 53, no. 1, pp. 109–115.
34. **Babu T.** et al. Particle swarm optimization based solar PV array reconfiguration of the maximum power extraction under partial shading conditions, *IEEE Trans. Sustain. Energy*, 2017, vol. 30, pp. 129–138.
35. **Prakash D., Lakshminarayana C.** Optimal siting of capacitors in radial distribution network using whale algorithm, *Alexandria Eng. J.*, 2016, no. 8, pp. 57–68.
36. **Massan S.** et al. Wind turbine micro siting by using the firefly algorithm, *Appl. Soft Comput. J.*, 2015, vol. 27, pp. 450–456.
37. **Mafarja M., Mirjalili S.** Hybrid whale optimization algorithm with simulated annealing for feature selection, *Neurocomputing*, 2016, vol. 260, pp. 302–312.
38. **Moayedikia A.** et al. Feature selection for high dimensional imbalanced class data using harmony search, *Eng. Appl. Artif. Intell.*, 2017, vol. 57, pp. 38–49.
39. **Alswaiti M., Albughdadi M.** Density-based particle swarm optimization algorithm for data clustering, *Expert Syst. Appl.*, 2018, vol. 91, pp. 170–186.
40. **Abualigah L.** et al. A novel hybridization strategy for krill herd algorithm applied to clustering techniques, *Appl. Soft Comput.*, 2017, vol. 60, pp. 423–435.
41. **Abualigah L., Khader A.** Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, *Supercomputing*, 2017, vol. 73(11), pp. 4773–4795.
42. **Li P., Duan H.** Path planning of unmanned aerial vehicle based on improved gravitational search algorithm, *Chin Technol. Sci.*, 2012, vol. 55(10), pp. 2712–2719.
43. **Oz I., Topcuoglu H.** A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles, *Simulation*, 2013, vol. 89(8), pp. 903–920.
44. **Silva P.** et al. Automatic generation of biped locomotion controllers using genetic programming, *Rob. Auton. Syst.*, 2014, vol. 62(10), pp. 1531–1548.
45. **Rodzin S. I., Skobtsov Yu. A., El-Khatib S. A.** Bio heuristics: Theory, Algorithms, and Applications, Cheboksary: ID Sreda, 2019, 224 p. (in Russian)
46. **Wang K.** et al. An improved electromagnetism-like mechanism algorithm and its application to the prediction of diabetes, *J. Biomed. Inform.*, 2015, vol. 54, pp. 220–229.
47. **Alshamlan H., Badr G., Alohali Y.** MRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling, *Biomed Res. Int.*, 2015, vol. 23, pp. 201–215.
48. **Saji Y., Riffi M.** A novel discrete algorithm for solving the travelling salesman problem, *Neural Comput. Appl.*, 2016, vol. 27(7), pp. 1853–1866.
49. **Wei L.** A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints, *Eur. J. Oper. Res.*, 2017, no. 5, pp. 1–17.
50. **Jourdan L., Basseur M., Talbi E.** Hybridizing exact methods and metaheuristics: a taxonomy, *Eur. J. Oper. Res.*, 2009, vol. 199(3), pp. 620–629.
51. **Kureichik V., Kuliev E., Kureichik V.** Monkey search algorithm for ECE components partitioning, *J. of Physics: Conf. Series*, 2018, vol. 1015 (4), no. UNSP 042026, pp. 1–10.
52. **Kureichik V., Kuliev E., Kureichik V.** Mechanisms of swarm intelligence and evolutionary adaptation for solving PCB design tasks, *Proc. Int. Sem. on Electron Devices Design and Production (SED)*, 2019, pp. 109–113.

В. Н. Гридин¹, науч. руководитель, д-р техн. наук, проф., e-mail: info@ditc.ras.ru,

В. И. Анисимов^{1, 2}, гл. науч. сотр., д-р техн. наук, проф., e-mail: info@ditc.ras.ru,

С. А. Васильев², канд. техн. наук, ассистент, e-mail: venom-gt@list.ru,

¹ Центр информационных технологий в проектировании РАН,

² Санкт-Петербургский государственный электротехнический университет

Методы расчета переходных процессов в больших электронных схемах на основе диакоптики*

Рассматривается методика построения математического и программного обеспечения для моделирования переходных процессов в больших электронных схемах на основе декомпозиции исходной схемы на ряд подсхем с последующим расчетом схемы по частям методами диакоптики. Практическая реализация предлагаемой методики позволяет значительно увеличить производительность программного обеспечения, снизить затраты требуемой оперативной памяти и увеличить скорость решения задачи расчета переходных процессов, что особенно актуально при разработке распределенных систем автоматизации схемотехнического проектирования вследствие уменьшения времени диалогового взаимодействия системы с сетью Интернет.

Ключевые слова: распределенные системы, автоматизация схемотехнического проектирования, диакоптика, моделирование схем, обобщенный сигнальный граф, методы декомпозиции, технология расчета слабосвязанных схем по частям

Введение

При расчете переходных процессов в больших электронных схемах часто возникает необходимость в использовании ресурсов, превосходящих имеющиеся в распоряжении вычислительные ресурсы. Если моделируемая схема имеет слабосвязанную структуру, то наиболее эффективным способом решения проблемы недостаточности ресурсов является декомпозиция исходной схемы на ряд подсхем на основе диакоптического подхода, использование которого существенно повышает эффективность вычислительных процессов [1–4]. Для возможности реализации диакоптического подхода к моделированию больших схем необходимо обеспечить отсутствие индуктивных связей между отдельными подсхемами, а также концентрацию внутри каждой подсхемы управляющих и управляемых переменных зависимых источников. Хотя в остальном способ декомпозиции системы на отдельные составляющие подсистемы является произвольным, с практической точки зрения целесообразно в качестве отдельных подсхем выбирать функциональные блоки, каждый из которых выполняет определенную задачу. Связь между отдельными подсхемами при декомпозиции схемы осуществляется через выделенные переменные, которые содержат все узловые напряжения узлов связи.

При диакоптическом подходе к расчету больших электронных схем не требуется фор-

мирования полной системы уравнений, достаточно сформировать уравнения для ее элементарных подсхем, которые могут быть сделаны настолько малыми, насколько это практически целесообразно. Общее решение задачи моделирования получается путем соединения полученных частных решений для подсхем, на которые разделена исходная схема.

Для описания моделируемой схемы на основе методов диакоптики предлагается осуществить топологическое описание схемы с использованием обобщенных сигнальных графов, отображающих матричные уравнения схемы, записанные в обобщенной причинно-следственной форме [5, 6]. Описание задачи на основе обобщенных сигнальных графов дает наглядное представление о структуре связей между подсхемами, а также позволяет простейшим способом реализовать выбор последовательности нумерации переменных узлов связи и внутренних переменных подсхем при декомпозиции исходной схемы.

Достоинства методов декомпозиции и расчета больших схем по частям оказываются весьма значительными особенно при анализе больших электронных схем, при этом сокращение затрат на время решения задачи тем больше, чем сложнее анализируемая схема. При этом возможна организация как последовательного вычислительного процесса, так и организация параллельных вычислений в сети, где компьютер каждого узла осуществляет формирование и обработку данных, связанных только с отдельной подсхемой. Возможность распараллеливания вычислительных процессов существенно ускоряет решение

*Работа выполняется в рамках темы № 0071-2019-0001.

задач и позволяет реализовать построение распределенных систем моделирования, функционирующих в сети Интернет [7–9].

Построение программного обеспечения для расчета переходных процессов в больших электронных схемах на основе их декомпозиции с последующим расчетом схемы по частям методами диакоптики может быть осуществлено на основе двух способов организации итерационных циклов — путем организации сквозного вычислительного процесса для всей схемы или на основе автономных вычислительных процессов для отдельных подсхем с последующим процессом расчета переменных связи. Показывается, что расчет динамических режимов на основе автономных вычислительных процессов для отдельных подсхем оказывается целесообразным в случаях, когда отдельные подсхемы имеют свойства, существенно отличные от свойств других подсхем.

Дискретизация и линеаризация компонентов

Формирование математического описания для расчета переходных процессов в схеме целесообразно выполнять на основе алгебро-дифференциального уравнения моделируемой системы, записанного в неявном виде:

$$F(\dot{X}, X, t) = 0.$$

Здесь X — вектор переменных, \dot{X} — вектор производных, t — текущее время.

Практическая реализация построения алгебро-дифференциальных уравнений может быть легко осуществлена путем формирования уравнений моделируемой схемы в расширенном базисе узловых потенциалов. Для этой цели следует включить все полюсы емкостных компонентов схемы в число u -полюсов, а все полюсы индуктивных компонентов — в число i -полюсов, выделив эти элементы из структуры многополюсников. При этом уравнение моделируемой можно записать в виде

$$(sD + W)X + S = 0,$$

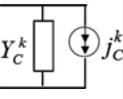
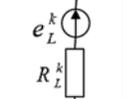
где D — диагональная блочная матрица емкостных и индуктивных компонентов схемы; W — матрица частотно-независимых компонентов схемы; X — вектор переменных; S — вектор задающих источников схемы.

Заменяя оператор s операцией дифференцирования, получим алгебро-дифференциальное уравнение моделируемой системы в неявной форме, которое для момента времени $t = k + 1$ имеет вид $F(\dot{X}^{k+1}, X^{k+1}, t^{k+1}) = 0$. Для исключения вектора производной \dot{X}^{k+1} следует воспользоваться известными формулами коррек-

ции $\dot{X}^{k+1} = aX^{k+1} + b$, где a, b — коэффициенты формул интегрирования для многошаговых неявных методов высших порядков Адамса—Маултона или Гира [10, 11], что приводит к уравнению $F(aX^{k+1} + b, X^{k+1}, t^{k+1}) = 0$, которое можно решить методом Ньютона—Рафсона относительно вектора X^{k+1} , поскольку значения коэффициентов a и b к моменту времени $t = k + 1$ уже определены на предыдущих шагах расчета. В качестве начального приближения $(X^{k+1})^i = (X^{k+1})^0$ при этом можно принять значение вектора на предыдущем шаге X^k . Для повышения точности расчета можно также воспользоваться прогнозируемым значением вектора $(X^{k+1})^0$, которое вычисляется на основании явных методов Адамса—Башфорта или Гира.

При реализации программного обеспечения процесс исключения значений производных практически выполняется не в результирующем матричном уравнении схемы, а непосредственно при формировании дискретизированных компонентных уравнений $i_C^{k+1} = Y_C^k u_C^{k+1} + j_C^k$ и $u_L^{k+1} = R_L^k i_L^{k+1} + e_L^k$, где i_C^{k+1}, u_C^{k+1} — токовая и потенциальная переменные на $(k + 1)$ -м шаге; Y_C^k, j_C^k — проводимость и значение источника тока на k -м шаге; u_L^{k+1}, i_L^{k+1} — потенциальная и токовая переменные на $(k + 1)$ -м шаге; R_L^k, e_L^k — сопротивление и значение источника напряжения на k -м шаге дискретных моделей для двухполюсников C и L , а также для частотно-зависимых управляемых источников.

В таблице приведены дискретные схемы замещения для двухполюсников C и L на основе методов Адамса—Маултона и Гира второго порядка ($m = 2$).

Компонент	Дискретная модель	$m = 2$ (метод Адамса-Маултона)	$m = 2$ (метод Гира)
		$Y_C^k = \frac{2C}{h^k}$ $j_C^k = -\frac{2C}{h^k} u_C^k - i_C^k$	$Y_C^k = \frac{3C}{2h^k}$ $j_C^k = -\frac{C}{2h^k} (4u_C^k - u_C^{k-1})$
		$R_L^k = \frac{2L}{h^k}$ $e_L^k = -\frac{2L}{h^k} i_L^k - u_L^k$	$R_L^k = \frac{3L}{2h^k}$ $e_L^k = -\frac{L}{2h^k} (4i_L^k - i_L^{k-1})$

В процессе решения алгебро-дифференциального уравнения $F(\dot{X}^{k+1}, X^{k+1}, t^{k+1}) = 0$ необходимо решать методом Ньютона—Рафсона уравнение $F(aX^{k+1} + b, X^{k+1}, t^{k+1}) = 0$, которое в общем случае является нелинейным. Решение нелинейного уравнения $\Phi(X) = 0$ методом Ньютона—Рафсона основано на использовании итерационного уравнения вида

$$M^i \Delta X^{i+1} + \Phi(X^i) = 0, \quad (1)$$

где $M^i = \frac{\partial \Phi(X)}{\partial X} \Big|_{X=X^i}$; $\Delta X^{i+1} = X^{i+1} - X^i$ — матрица Якоби в точке $X = X^i$.

Недостатком общей методики является необходимость вычисления на каждом шаге итерационного процесса функции $\Phi(X^i)$ и матрицы производных M^i , что существенно затрудняет использование классического алгоритма при моделировании больших систем. Поэтому следует использовать схемотехническую интерпретацию метода Ньютона—Рафсона, основанную на линейаризации нелинейных компонент схемы в каждой точке итерационного процесса.

Так, если нелинейные компоненты в общем случае описываются матричным уравнением $P_2 = F(Q_2)$, то разлагая это уравнение в ряд Тейлора в точке $Q_2 = Q_2^i$ и отбрасывая высшие производные, получим уравнение вида

$$P_2 = W_2^i Q_2 + S_2^i, \quad (2)$$

где $W_2^i = \frac{\partial P_2}{\partial Q_2} \Big|_{Q_2^i}$, $S_2^i = P_2 - W_2^i Q_2^i$ — линейаризованные в точке $Q_2 = Q_2^i$, $P_2 = P_2^i$ матрицы неавтономных и автономных параметров нелинейных компонент.

Пусть нелинейная схема описывается уравнением

$$\Phi(X) = W_1 X + S_1 + S_2 = 0, \quad (3)$$

где W_1 и S_1 — матрица и задающий вектор линейной части схемы; $S_2 = P_2 = F(Q_2)$ — вектор фиктивных задающих источников, отображающих нелинейные характеристики компонент.

Тогда на основании уравнения линейаризованных компонент (2) и уравнения нелинейной схемы (3) можно привести уравнение Ньютона—Рафсона (1) к виду [12]

$$(W_1 + W_2^i) X^{i+1} + S_1 + S_2^i = 0, \quad (4)$$

где W_1 и S_1 — матрица и задающий вектор, описывающие линейную часть схемы; W_2^i и S_2^i — матрица и задающий вектор линейаризованной в точке $Q_2 = Q_2^i$ нелинейной части схемы.

Построение итерационного вычислительного процесса на основе схемотехнической интерпретации метода Ньютона—Рафсона в соответствии с уравнением (4) не требует построения матрицы Якоби, так как для решения задачи достаточно провести линейаризацию нелинейных компонент в очередной точке итерационного процесса.

Практическую реализацию схемотехнической интерпретации удобнее выполнять на ос-

нове линейаризации отдельных нелинейных компонент. Так, если некоторая компонента имеет нелинейные характеристики вида $p_i = \varphi_i(\dots, q_j, \dots)$, $i = \overline{1, n}$, $j = \overline{1, n}$, то после разложения характеристик в ряд Тейлора и отбрасывания высших производных получим линейаризованные уравнения компонент

$$p_i = \sum_{j=1}^n w_{ij}^i q_j + s_i^i, \quad i = \overline{1, n}, \quad (5)$$

где $w_{ij}^i = \frac{\partial \varphi_i}{\partial q_j}$, $s_i^i = p_i - \sum_{j=1}^n w_{ij}^i q_j^i$ — неавтономные и автономные линейаризованные параметры компонент схемы.

На основании уравнения компонент (5) трудно сформировать линейаризованные уравнения (4), которые лежат в основе схемотехнической интерпретации метода Ньютона—Рафсона.

Топологическое описание задачи расчета динамических режимов большой схемы по частям

В результате дискретизации частотно-зависимых компонент и линейаризации нелинейных многополюсников математическое обеспечение моделируемой схемы может быть представлено на каждом шаге соответствующих итерационных процессов линейными алгебраическими уравнениями. Хотя решение этих уравнений может быть выполнено стандартными методами, при моделировании больших систем часто возникают задачи, для которых требуется использовать вычислительные ресурсы, превосходящие реальные возможности компьютера. Эффективным способом решения этой проблемы является декомпозиция моделируемой схемы на ряд слабосвязанных подсхем с последующим расчетом схемы по частям на основе методов диакоптики. В результате проведенной декомпозиции появляется возможность существенного увеличения эффективности использования оперативной памяти, а также возрастает скорость решения задачи расчета больших схем. Практическая реализация декомпозиции схемы на отдельные составляющие подсхемы может быть произвольной, однако целесообразно в процессе декомпозиции оформлять в виде отдельных подсхем функциональные блоки, решающие некоторые законченные задачи. При декомпозиции схемы необходимо только обеспечивать отсутствие индуктивных связей между отдельными подсхемами и локализацию внутри каждой подсхемы ее зависимых источников, а также управляющих переменных. Связь между отдельными подсхемами осуществляется путем введения перемен-

ных, которые содержат все узловые напряжения узлов связи подсхем.

Для описания моделируемой схемы на основании декомпозиционного подхода целесообразно использовать обобщенные сигнальные графы, которые в отличие от сигнальных графов Мэзона [5] отображают системы уравнений, записанные в обобщенной причинно-следственной форме, содержащей взвешенные вершины [6]. Помимо наглядного представления о структуре связей между подсхемами такое топологическое описание позволяет устранить проблему выбора последовательности нумерации переменных, выделенных для узлов связи, и внутренних переменных отдельных подсхем.

Для топологического описания моделируемой схемы на основе обобщенного сигнального графа образуем после декомпозиции схемы вектор внутренних переменных каждой подсхемы $X_m = [\dots, x_i, \dots]^T$ и вектор переменных узлов связи подсхем $X_0 = V_0 = [\dots, v_p, \dots]^T$. Тогда можно построить уравнение k -й подсхемы на j -м шаге дискретизации и i -м шаге линеаризации в обобщенной причинно-следственной форме:

$$(W_{kk}^j)^i (X_k^j)^i = -(W_{k0}^j)^i (X_0^j)^i - (S_k^j)^i = 0, \quad (6)$$

$$k = \overline{1, m},$$

где $(W_{kk}^j)^i$, $(W_{k0}^j)^i$ — матрицы параметров; $(S_k^j)^i$ — вектор задающих источников k -й подсхемы на j -м шаге дискретизации и i -м шаге линеаризации.

Уравнение для переменных узлов связи на j -м шаге дискретизации и i -м шаге линеаризации в обобщенной причинно-следственной форме будет иметь вид

$$(W_{00}^j)^i (X_0^j)^i = -\sum_{k=1}^m (W_{0k}^j)^i (X_k^j)^i - (S_0^j)^i, \quad (7)$$

где $(W_{00}^j)^i$, $(W_{0k}^j)^i$ — матрицы параметров; $(S_0^j)^i$ — вектор задающих источников узлов связи на j -м шаге дискретизации и i -м шаге линеаризации.

На основании уравнений (6) и (7) можно построить топологическое описание схемы на основе обобщенного сигнального графа (рис. 1). Множества вершин $(X_k^j)^i$, $(S_k^j)^i$, $(X_0^j)^i$, $(S_0^j)^i$ ($k = \overline{1, m}$) обобщенного сигнального графа сопоставлены соответствующим векторам $(X_k^j)^i$, $(S_k^j)^i$, $(X_0^j)^i$, $(S_0^j)^i$, а множество весов ветвей и вершин этого графа $(W_{kk}^j)^i$, $(W_{k0}^j)^i$, $(W_{0k}^j)^i$, сопоставлены соответствующим матрицам $(W_{kk}^j)^i$, $(W_{k0}^j)^i$, $(W_{0k}^j)^i$, $(W_{00}^j)^i$ уравнений системы (6) и (7).

Построенный обобщенный сигнальный граф отображает взаимодействие между отдельными подсхемами, которое обеспечивается через пере-

менные узлы связи. При этом легко реализуется независимая обработка описания подсхем и узлов связи.

Удаляя в топологическом описании каждой подсхемы все ветви, заходящие в вершины и исходящие из вершин множества $(X_k^j)^i$ с последующей нормализацией к единичным значениям весовых коэффициентов взвешенных вершин каждой подсхемы, получим преобразованный граф, приведенный на рис. 2.

Для преобразованного графа уравнения подсхем имеют вид

$$(X_k^j)^i = -(\bar{W}_{k0}^j)^i (X_0^j)^i - (\bar{S}_k^j)^i. \quad (8)$$

Значения составляющих множества весов $(\bar{W}_{k0}^j)^i$ и $(\bar{S}_k^j)^i$ определяются на основании топологических формул пересчета веса ветвей графа при нормализации весовых коэффициентов вершин множества $(X_k^j)^i$ [6]. В матричной форме указанные топологические преобразования соответствуют алгоритму Гаусса—Жордана приведения описания к диагональной матрице с единичными значениями диагональных коэффициентов.

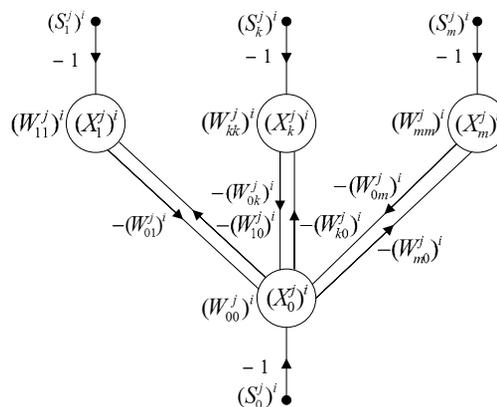


Рис. 1. Обобщенный сигнальный граф моделируемой схемы

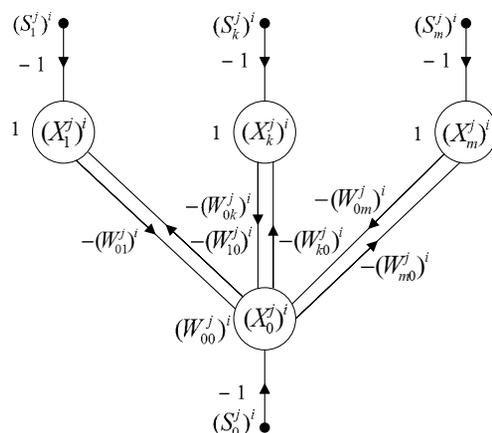


Рис. 2. Преобразованный обобщенный сигнальный граф

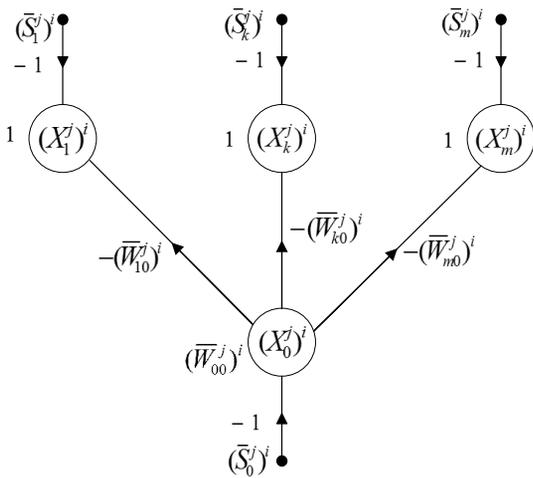


Рис. 3. Результирующий обобщенный сигнальный граф

Подставляя уравнение (8) в уравнение (7), получим уравнение для расчета вектора $(X_0^j)^i$ переменных связи:

$$(\bar{W}_{00}^j)^i (X_0^j)^i = -(\bar{S}_0^j)^i, \quad (9)$$

где

$$(\bar{W}_{00}^j)^i = (W_{00}^j)^i - \sum_{k=1}^m (W_{0k}^j)^i (\bar{W}_{k0}^j)^i;$$

$$(\bar{S}_0^j)^i = (S_0^j)^i - \sum_{k=1}^m (W_{0k}^j)^i (\bar{S}_k^j)^i.$$

Уравнению (9) соответствует результирующий обобщенный сигнальный граф, приведенный на рис. 3.

На основании полученной топологической структуры можно рассчитать вектор переменных связи $X_0 = V_0 = [\dots, v_p, \dots]^t$, после чего найти значения векторов внутренних переменных всех подсхем $X_k = [\dots, x_i, \dots]^t$.

Методы организации итерационных циклов

Для построения алгоритмов расчета переходных процессов в больших электронных схемах методами диакоптики введем векторы дифференцируемых токов I_s и дифференцируемых напряжений U_s и образуем вектор дифференцируемых переменных $Q_s = [U_s^t, I_s^t]^t$ нелинейных компонентов схемы. Выберем такой расширенный базис узловых потенциалов, вектор токов которого включает в себя все составляющие вектора дифференцируемых токов I_s , что позволяет определить все составляющие вектора дифференцируемых переменных через составляющие координатного базиса, при этом потенциальные переменные определяются на основании соотношения $U_s = A_s^t V$. Введем также векторы Q_f и P_f для переменных, входящих в функциональные зависимости нелинейных компонентов $P_f = F(Q_f)$.

Организация итерационных процессов при моделировании переходных процессов в больших электронных схемах на основе их декомпозиции с последующим расчетом схемы по частям методами диакоптики может быть осуществлена двумя способами — путем организации сквозного вычислительного процесса для всей схемы или на основе автономных вычислительных процессов для отдельных подсхем с последующим процессом расчета переменных связи.

Сквозной вычислительный процесс является наиболее общим типом организации вычислительного процесса и может быть практически реализован в соответствии со следующим алгоритмом:

1. Задание начальных значений $t^j = t^0$, $(X_k^j)^i = X_k^0$, $(X_0^j)^i = X_0^0$.
 2. $t^{j+1} = t^j + h$.
 3. Начало 1-го цикла $k = \overline{1, m}$.
 4. Дискретизация частотно-зависимых компонент в точке Q_s^j .
 5. Установка начального значения для линеаризации $(X_k^{j+1})^i = X_{\text{пр}}^j$.
 6. Линеаризация нелинейных компонент в точке $(Q_f^{j+1})^i$.
 7. Формирование блочных матриц $(W_{kk}^j)^i$, $(W_{k0}^j)^i$, $(W_{0k}^j)^i$, $(W_{00}^j)^i$ и векторов $(S_k^j)^i$, $(S_0^j)^i$ подсхем.
 8. Преобразование $(W_{kk}^j)^i$ к единичной матрице и расчет $(\bar{W}_{k0}^j)^i$, $(\bar{S}_k^j)^i$.
 9. Расчет поправок $(\Delta_k W_{00}^j)^i = (W_{0k}^j)^i (\bar{W}_{k0}^j)^i$, $(\Delta_k S_0^j)^i = (W_{0k}^j)^i (\bar{S}_k^j)^i$ и вычисление модифицированных значений блочной матрицы и задающего вектора узлов связи
- $$(\bar{W}_{00}^j)^i = (W_{00}^j)^i - \sum_{k=1}^m (\Delta_k W_{00}^j)^i$$
- $$(\bar{S}_0^j)^i = (S_0^j)^i - \sum_{k=1}^m (\Delta_k S_0^j)^i$$
10. Окончание 1-го цикла $k = \overline{1, m}$.
 11. Решение уравнения $(\bar{W}_{00}^j)^i (X_0^{j+1})^{i+1} + (\bar{S}_0^j)^i = 0$ для переменных связи.
 12. Начало 2-го цикла $k = \overline{1, m}$.
 13. Расчет вектора внутренних переменных подсхем $(X_k^{j+1})^{i+1} = -(\bar{W}_{k0}^j)^i (X_0^{j+1})^{i+1} - (\bar{S}_k^j)^i$.
 14. Окончание 2-го цикла $k = \overline{1, m}$.
 15. Расчет нормы $N = \|(X_k^{j+1})^{i+1} - (X_k^{j+1})^i\|$.
 16. Если $N > \varepsilon$, то переопределение векторов внутренних переменных подсхем $(X_k^{j+1})^i = (X_k^{j+1})^{i+1}$ и вектора переменных связи $(X_0^{j+1})^i = (X_0^{j+1})^{i+1}$ и возврат к п. 5, иначе вывод векторов внутренних переменных подсхем $(X_k^{j+1})^{i+1}$ и переменных связи $(X_0^{j+1})^{i+1}$.
 17. Если $t^{j+1} < t_{\text{оконч}}$, то переход к п. 2, иначе конец расчета.

Построение программного обеспечения для расчета динамических режимов на основе автономных вычислительных процессов для отдельных подсхем может оказаться целесообразным в случаях, когда отдельные подсхемы имеют свойства, существенно отличные от свойств других подсхем. К таким ситуациям относится случай, когда степень нелинейности отдельных подсистем значительно отличается от степени нелинейности других подсхем, например, когда все нелинейные свойства схемы можно сконцентрировать в одной из подсхем, а все остальные подсхемы содержат только линейные элементы. Этот случай организации автономных вычислительных процессов рассмотрен в работе [12] и здесь не рассматривается. Другим случаем неоднородности свойств подсхем является случай, когда моделируемая схема содержит "быстрые" и "медленные" подсхемы, существенно отличающиеся своей инерционностью. Сущность методики организации автономного вычислительного процесса при расчете динамических режимов схем, содержащих подсхемы, значительно отличающиеся своей инерционностью, заключается в том, что для каждой подсхемы выбирается свой шаг расчета h_k , и с этим шагом выполняется расчет вектора внутренних переменных подсхем X_k^{j+1} . При этом устанавливаются точки синхронизации с шагом $H > h_k$, в которых расчет процессов в подсхемах прекращается и начинается расчет вектора переменных связи X_0^{j+1} с последующим уточнением вектора внутренних переменных X_k^{j+1} .

Алгоритм расчета динамических режимов электронных схем на основе декомпозиции и построения автономных вычислительных процессов может быть представлен в следующем виде:

1. Задание начальных значений $T^j = T^0$, $X_k^j = X_k^0$, $X_0^j = X_0^0$.
2. Начало 1-го цикла $k = \overline{1, m}$.
3. $t^j = T^j$.
4. $t^{j+1} = t^j + h_k$.
5. Дискретизация частотно-зависимых компонентов в точке Q_s^j .
6. Формирование блочных матриц $W_{kk}^j, W_{k0}^j, W_{0k}^j, W_{00}^j$ и векторов S_k^j, S_0^j подсхем.
7. Преобразование W_{kk}^j к единичной матрице и расчет $(\overline{W}_{k0}^j)^i, (\overline{S}_k^j)^i$.
8. Расчет вектора внутренних переменных подсхем $X_k^{j+1} = -\overline{W}_{k0}^j X_0^j - \overline{S}_k^j$.
9. Если $t^{j+1} \leq H$, то переопределение $t^j = t^{j+1}$, $X_k^j = X_k^{j+1}$ и возврат к п. 4, иначе переход к п. 10.
10. Расчет поправок $\Delta_k W_{00}^j = W_{0k}^j \overline{W}_{k0}^j$, $\Delta_k S_0^j = W_{0k}^j \overline{S}_k^j$ и вычисление модифицированных значений блочной матрицы и задающего вектора узлов связи $\overline{W}_{00}^j = W_{00}^j - \sum_{k=1}^m \Delta_k W_{00}^j$; $\overline{S}_0^j = S_0^j - \sum_{k=1}^m \Delta_k S_0^j$.

11. Окончание 1-го цикла $k = \overline{1, m}$.
12. $t^{j+1} = T^j + H$.
13. Решение уравнения $\overline{W}_{00}^j X_0^{j+1} + \overline{S}_0^j = 0$ для переменных связи.
14. Начало 2-го цикла $k = \overline{1, m}$.
15. Уточнение вектора внутренних переменных подсхем $X_k^{j+1} = -\overline{W}_{k0}^j X_0^{j+1} - \overline{S}_k^j$.
16. Окончание 2-го цикла $k = \overline{1, m}$.
17. Вывод $t^{j+1}, X_k^{j+1}, X_0^{j+1}$.
18. Если $t^{j+1} < t_{\text{оконч}}$, то переопределение $T^j = t^{j+1}$, $X_k^j = X_k^{j+1}$, $X_0^j = X_0^{j+1}$ и возврат к п. 2, иначе конец расчета.

Использование автономных вычислительных процессов дает существенный выигрыш при расчете схем, в которых можно выделить "быстрые" и "медленные" подсхемы, поскольку для расчета "быстрых" подсхем можно выбрать достаточно малые шаги расчета h_k . Пусть, например, моделируемая схема содержит две подсхемы — "быструю", которую можно рассчитывать с шагом $h_1 = 10^{-6}$ с, и "медленную", которую необходимо рассчитывать с шагом $H = h_2 = 1$ с. Если $t_{\text{оконч}} = 10$ с, и для расчета подсхем и переменных связи на одном шаге необходимо выполнить N_{BO} вычислительных операций, то в этом случае число вычислительных операций при автономном вычислительном процессе за время $t_{\text{оконч}} = 10$ с составит

$$N_{\text{BO1}} = (10^6 + 1 + 1) \cdot 10 N_{\text{BO}} = 10^7 N_{\text{BO}}$$

При сквозном вычислительном процессе имеем

$$N_{\text{BO2}} = 3 \cdot 10^6 \cdot 10 N_{\text{BO}} = 3 \cdot 10^7 N_{\text{BO}}$$

Таким образом, при организации автономного вычислительного процесса время решения задачи будет в три раза меньше, чем при сквозном процессе вычислений.

Заключение

Предложена методика моделирования переходных процессов в больших электронных схемах на основе методов диакоптики путем декомпозиции исходной схемы на ряд подсхем с последующим расчетом схемы по частям. Описание моделируемой схемы предлагается осуществлять на основе обобщенных сигнальных графов, отображающих матричные уравнения схемы, записанные в обобщенной причинно-следственной форме, что обеспечивает наглядное представление о структуре иерархических связей между отдельными подсхемами и предоставляет возможность реализовать выбор произвольной последовательности нумерации переменных для узлов связи и внутренних переменных подсхем. Рассмотрены способы организации итераци-

онных циклов при моделировании переходных процессов в больших электронных схемах методами диакоптики путем организации сквозного вычислительного процесса для всей схемы и на основе автономных вычислительных процессов для отдельных подсхем.

Реализация предложенной методики расчета переходных процессов в больших электронных схемах не требует предварительного формирования полного описания всей схемы, достаточно последовательно сформировать уравнения для ее элементарных подсхем, что позволяет существенно снизить затраты требуемой оперативной памяти и увеличить скорость решения задачи моделирования.

Список литературы

1. Крон Г. Исследование сложных схем по частям — диакоптика / Пер. с англ. М.: Наука, 1972. 542 с.
2. Хэпп Х. Диакоптика и электрические цепи / Пер. с англ. М.: Мир, 1974. 342 с.

3. Баталов Б. В., Егоров Ю. Б., Русаков С. Г. Основы математического моделирования больших интегральных схем на ЭВМ. М.: Радио и связь, 1982. 168 с.
4. Анисимов В. И., Тарасова О. Б., Алмаасали С. А. Организация вычислительных процессов при моделировании схем на основе методов диакоптики // Информационные технологии в проектировании и производстве. 2013. № 4. С. 14—17.
5. Мэзон С., Циммерман Г. Электронные цепи, сигналы и схемы / Пер. с англ. М.: Изд. иностр. лит., 1963. 619 с.
6. Анисимов В. И. Топологический расчет электронных схем. Л.: Энергия, 1977. 238 с.
7. Коваленко О. С., Курейчик В. М. Обзор проблем и состояний облачных вычислений и сервисов // Известия ЮФУ. Технические науки. 2012. № 7. С. 146—153.
8. Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А. Построение схем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. № 5. С. 23—27.
9. Анисимов Д. А. Методы построения схем автоматизации схемотехнического проектирования на основе веб-сервисов // Известия СПбГЭТУ "ЛЭТИ". 2012. № 10. С. 56—61.
10. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем / Пер. с англ. М.: Радио и связь, 1988. 560 с.
11. Чуа Л. О., Лин П. М. Машинный анализ электронных схем / Пер. с англ. М.: Энергия, 1980. 832 с.
12. Гридин В. Н., Анисимов В. И. Моделирование нелинейных систем на основе методов декомпозиции // Информационные технологии. 2020. Т. 26, № 3. С. 131—137.

V. N. Gridin¹, Scientific Director, Dr. Sci. Tech., Professor, e-mail: info@ditc.ras.ru,
V. I. Anisimov^{1,2}, Chief Researcher, Dr. Sci. Tech., Professor, e-mail: info@ditc.ras.ru,
S. A. Vasilev², PhD, Assistant, e-mail: venom-gt@list.ru

¹ Design Information Technologies Center Russian Academy of Sciences,
Odintsovo, Moscow region, Russian Federation,

² Saint Petersburg Electrotechnical University, Saint Petersburg, Russian Federation

Methods for Calculating Transients in Large Electronic Circuits Based on Diacoptics

The article deals with the method of constructing mathematical and software for modeling transients in large electronic circuits based on the decomposition of the original circuit into a number of subcircuits, followed by the calculation of the circuit in parts by diacoptics methods. The practical implementation of the proposed method allows you to significantly increase the performance of the software, reduce the cost of the required RAM and increase the speed of solving the problem of calculating transients, which is especially important in the development of distributed automation systems for circuit design due to the reduced time of dialog interaction between the system and the Internet.

Keyword: distributed systems, automation of circuit design, diacoptics, circuit modeling, generalized signal graph, decomposition methods, technology for calculating weakly connected circuits in parts

DOI: 10.17587/it.27.575-581

References

1. Kron G. Investigation of complex schemes in parts diacoptics, Moscow, Nauka, 1972, 542 p.
2. Happ H. Diacoptics and electrical circuits, Moscow, Mir, 1974, 342 p.
3. Batalov B. V., Egorov Yu. B., Rusakov S. G. Fundamentals of mathematical modeling of large-scale integrated circuits on a computer, Moscow, Radio and communication, 1982, 168 p.
4. Anisimov V. I., Tarasova O. B., Almaasali S. A. Organization of computational processes in modeling circuits based on diacoptics methods, *Information technologies in design and production*, 2013, no. 4, pp. 14—17.
5. Mason S., Zimmerman G. Electronic circuits, signals and circuits, Moscow, Izd. Inostr. Literatiry, 1963, 619 p.
6. Anisimov V. I. Topological calculation of electronic circuits, Leningrad, Energy, 1977, 238 p.

7. Kovalenko O. S., Kureichik V. M. Review of problems and states of cloud computing and services, *Izvestia SFedU. Tekhnicheskie nauki*, 2012, no. 7, pp. 146—153.
8. Gridin V. N., Dmitreievich G. D., Anisimov D. A. Construction of computer-aided design schemes based on Web-technologies, *Informatsionnye tekhnologii*, 2011, no. 5, pp. 23—27.
9. Anisimov D. A. Methods for constructing automation schemes for circuit design based on web services, *Izvestiya SPb-GETU "LETI"*, 2012, no. 10, pp. 56—61.
10. Vlah I., Singhal K. Machine methods of analysis and design of electronic circuits, Moscow, Radio i svyaz, 1988, 560 p.
11. Chua L. O., Lin P. M. Machine analysis of electronic circuits, Moscow, Energiya, 1980, 832 p.
12. Gridin V. N., Anisimov V. I. Modeling of nonlinear systems based on decomposition methods, *Informatsionnye tekhnologii*, 2020, vol. 26, no. 3, pp. 131—137.

А. А. Сорокин, канд. техн. наук, доц., e-mail: alsorokin.astu@mail.ru,
Астраханский государственный технический университет

Исследование иерархических систем нечеткого вывода при получении интегральных оценок анализируемых объектов

Исследованы особенности формирования выходных закономерностей иерархическими системами нечеткого вывода (ИСНВ), основанными на СНВ, которые используют алгоритмы Мамдани или Такаги—Сугено. В результате показано, что значения выходных переменных исследуемых ИСНВ имеют свойство группироваться в области фиксированных величин. Выявленное свойство позволяет с помощью ИСНВ провести распределение объектов анализируемой выборки по группам состояний.

Ключевые слова: обработка информации, иерархическая система нечеткого вывода, алгоритм Мамдани, алгоритм Такаги—Сугено, функция принадлежности, распределение по группам, экспертная группа

Введение

Использование систем нечеткого вывода (СНВ) получило распространение в различных областях [1–5]. Одним из классов задач, для решения которых используются СНВ, является получение интегральной оценки на основе агрегирования входных переменных. В работах [6–10] СНВ рассматриваются как средство универсального аппроксимирования переменных, представленных в четкой и нечеткой формах описания. Следствием подобных возможностей СНВ стала реализация методов, позволяющих создавать самообучающиеся системы на основе теории нейронных сетей [10–13] или других эвристических методов [14]. Оператор для агрегирования переменных в подобных системах реализуется на основании анализа закономерностей в обучающей выборке. Одним из ограничений методов, основанных на анализе закономерностей в массивах данных, является требование по репрезентативности выборки, используемой для настройки вычислительной системы [15–17]. Применение экспертных знаний позволяет восполнить подобное ограничение. Для реализации экспертных систем также получили широкое распространение положения теории нечетких множеств [18–20]. В подобных системах эксперты идентифицируют оператор обработки информации в виде совокупности правил. Ограничением подобных СНВ является

"проклятие размерности", связанное с ростом числа правил при увеличении числа агрегируемых переменных [10, 11]. В работах [11, 21–24] предлагается разрешение "проклятия размерности" на основе использования иерархических СНВ (ИСНВ). Процедуры, заложенные в основу функционирования СНВ, формируют специфическую закономерность изменения значений выходной переменной в зависимости от значений входных переменных [10, 11, 25]. Подобные свойства при формировании ИСНВ приводят к искажениям результатов относительно эквивалентных им не иерархических СНВ.

Обзор литературы и постановка задачи

Как показывает обзор литературы [10, 11, 25], наиболее широкое распространение получили СНВ, в основу которых положены алгоритм Мамдани и алгоритм Такаги—Сугено. Функционирование этих алгоритмов заключается в следующем: пусть для описания входных переменных используется множество $X = \{x_j\}$, где $j = \overline{1, i_{\max}}$, а для описания выходных переменных используется множество $Y = \{y_j\}$, где $j = \overline{1, j_{\max}}$; каждому элементу множеств X и Y соответствует множество термов $T_x = \{t_{k_x}^x\}$, где $k_x = \overline{1, k_{x\max}}$, и $T_y = \{t_{k_y}^y\}$, где $k_y = \overline{1, k_{y\max}}$; каждому терму из множеств T_x и T_y соответствует функция принадлежности (ФП) $\mu_{k_x}^x$ и $\mu_{k_y}^y$;

кроме этого для каждого из элементов множеств X и Y существуют шкалы для описания значений в четкой sc_x^{cl} и нечеткой sc_x^{fz} формах представления. ФП $\mu_{k_x}^x$ используются для преобразования четких значений переменных x_i в нечеткие x_i^{fz} :

$$x_i^{fz} = \mu_{k_x}^x(x_i); x_i \in sc_x^{cl}, x_i^{fz} \in sc_x^{fz}; \quad (1)$$

взаимодействие различных значений переменных x_i^{fz} описывается с помощью правил. В рамках исследований рассматривались СНВ, которые используют правила со структурой MISO (multiple input single out):

$$p_n = \{[\mathcal{F}_{i=1}^{i_{\max}}(x_i^{fz} = t_{k_x}^x) \Rightarrow (y_j^{fz} = t_{k_y}^y)], \xi_n\}, \quad (2)$$

где $\mathcal{F}_{i=1}^{i_{\max}}$ — оператор агрегирования входных переменных, позволяющий на основании обобщения значений ФП входных переменных получить промежуточное значение ФП выходной переменной y_j^{fz} ; ξ_n — коэффициент доверия правила; n — номер правила. Для реализации $\mathcal{F}_{i=1}^{i_{\max}}$ используются логические операции "И", "ИЛИ", "НЕ". Совокупность правил (2) объединяются в базу знаний СНВ.

"Проклятие размерности" связано с увеличением числа правил при увеличении мощности терм-множеств входных переменных и числа самих входных переменных. При допущении, что мощность терм-множеств входных переменных одинакова, а при формировании базы знаний используется полный набор комбинаций термов, число правил n_{\max} определяется как

$$n_{\max} = k_{x, \max}^{i_{\max}}. \quad (3)$$

Для снятия ограничения в работах [11, 21–23] предлагается разделение множества входных переменных на несколько подмножеств:

$$X \subset X_1^1 \cup \dots \cup X_{u_1}^1; u_1 = \overline{1, u_{1\max}}, \quad (4)$$

для каждого подмножества X_u формируется СНВ. Результатом функционирования этих СНВ является формирование множества промежуточных выходных переменных первого уровня иерархии $Y^1 = \{y_u^1\}$. Множество Y^1 образует множество входных переменных следующего уровня иерархии, обозначим его X^2 (т.е. $Y^1 = X^2$). Аналогично (4) множество X^2 разделяется на некоторое число непересекающихся подмножеств:

$$X^2 \subset X_1^2 \cup \dots \cup X_{u_2}^2; u_2 = \overline{1, u_{2\max}}, \quad (5)$$

после чего переменные из выделенных подмножеств агрегируются СНВ второго уровня

иерархии. Операции продолжаются до идентификации множества целевых переменных Y_{end} . Число уровней в ИСНВ обозначается α . ИСНВ можно представить в виде ориентированного древовидного графа $G_{\text{ИСНВ}}(V_{\text{ИСНВ}}, E_{\text{ИСНВ}})$, где в состав множества вершин $V_{\text{ИСНВ}}$ входят элементы множества $X = V_{\text{лист}}$ — листья графа, множество промежуточных СНВ (ПСНВ) $V_{\text{ПСНВ}}$ — внутренние вершины графа, множество $V_{\text{корень}} = Y_{\text{end}}$ — корень графа, ребра графа E — соединения между листьями, вершинами и корнем графа.

Практика и анализ [10, 11, 25] показывают, что СНВ обладают неравномерной чувствительностью, которая выражается в том, что приращение значения выходной переменной может достаточно долго не изменяться при наличии приращений у множества входных переменных, а затем может наблюдаться резкий скачок значения выходной переменной при наличии незначительного приращения одной из входных переменных. Объединение СНВ в иерархическую структуру может усилить нежелательные искажения. Анализ работ [11, 21–23] показал, что вопрос исследования формирования выходных закономерностей ИСНВ от числа иерархий и типа алгоритма работы СНВ требует дальнейшего рассмотрения.

Цель данной работы заключается в исследовании закономерностей по формированию выходных значений в иерархических системах нечеткого вывода.

Исследование закономерностей функционирования иерархических систем нечеткого вывода

В работе под ИСНВ-М понимается вычислительная система, основанная на СНВ, использующих алгоритм Мамдани, под ИСНВ-ТС — вычислительная система, основанная на СНВ, использующих алгоритм Такаги—Сугено. К допущениям эксперимента относятся: (i) диапазон значений входных и выходных переменных изменяется от 0 до 100 баллов; (ii) входные переменные имеют положительную корреляцию с выходной переменной; (iii) в состав множества $V_{\text{ПСНВ}}$ входят однотипные СНВ, основанные на правилах, в которых значение терма выходной переменной равно минимальному семантическому значению термов входных переменных.

В ходе исследований проведены две серии экспериментов. В рамках первой серии исследо-

вали ИСНВ-М, в рамках второй серии анализировали ИСНВ-ТС. Каждая серия содержала две части. В первой части каждой из серий рассматривали СНВ, которые для описания переменных используют пять термов. Терм-множество для описания переменных с помощью пяти термов имело вид: $T_5 = \{t_1 = NB, t_2 = NM, t_3 = Z, t_4 = PM, t_5 = PB\}$, где NB означает понятие "низкое" значение, NM — понятие "между средним и низким" значениями, Z — понятие "среднее значение", PM — понятие "между высоким и средним" значениями, PB — понятие "высокое" значение.

В рамках второй части каждой из серий терм-множество для описания переменных с помощью девяти термов имело вид: $T_9 = \{t_1 = NB, t_2 = NM1, t_3 = NM2, t_4 = NM3, t_5 = Z, t_6 = PM1, t_7 = PM2, t_8 = PM3, t_9 = PB\}$, где $NM1$ — означает понятие "ближе к низкому, чем к среднему", $NM2$ — "между средним и низким", $NM3$ — "ближе к среднему, чем к низкому", $PM1$ — "ближе к среднему, чем к высокому", $PM2$ — "между средним и высоким", $PM3$ — "ближе к высокому, чем к среднему", остальные семантические значения термов аналогичны T_5 . Трапецидальные ФП используются для

описания термов входных и выходных переменных в СНВ на основе алгоритма Мамдани и для описания термов входных переменных в СНВ на основе алгоритма Такаги—Сугено, что показано на рис. 1.

Поверхности СНВ использующие для описания переменных по пять термов, показаны на рис. 2. Поверхности СНВ, которые используют по девять термов, имеют аналогичный характер.

Выходные переменные для СНВ на основе алгоритма Такаги—Сугено задаются с помощью констант. В случае использования для описания входных переменных терм-множества вида T_5 выходная переменная описывается терм-множеством:

$$T_5^y = \{t_1^y = NB, t_2^y = NM, t_3^y = Z, t_4^y = PM, t_5^y = PB\},$$

где $NB \sim 0, NM \sim 25, Z \sim 50, PM \sim 75, PB \sim 100$ баллов; при описании входных переменных терм-множества вида T_9 выходная переменная описывается терм-множеством:

$$T_9^y = \{t_1^y = NB, t_2^y = NM1, t_3^y = NM2, t_4^y = NM3, t_5^y = Z, t_6^y = PM1, t_7^y = PM2, t_8^y = PM3, t_9^y = PB\},$$

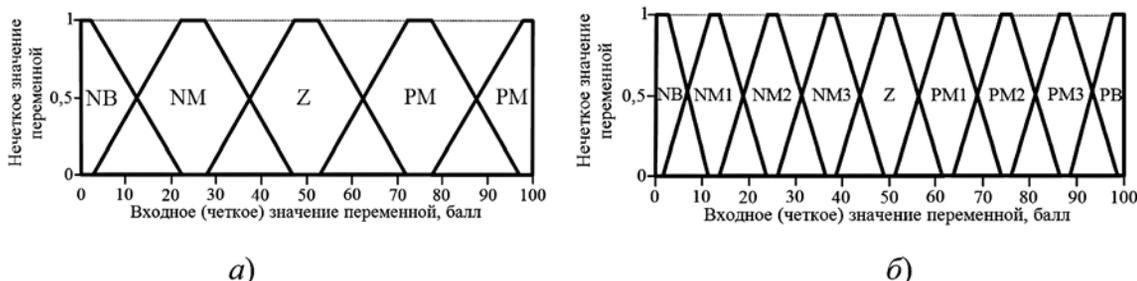


Рис. 1. Трапецидальные ФП для терм-множеств T_5 (а) и T_9 (б)

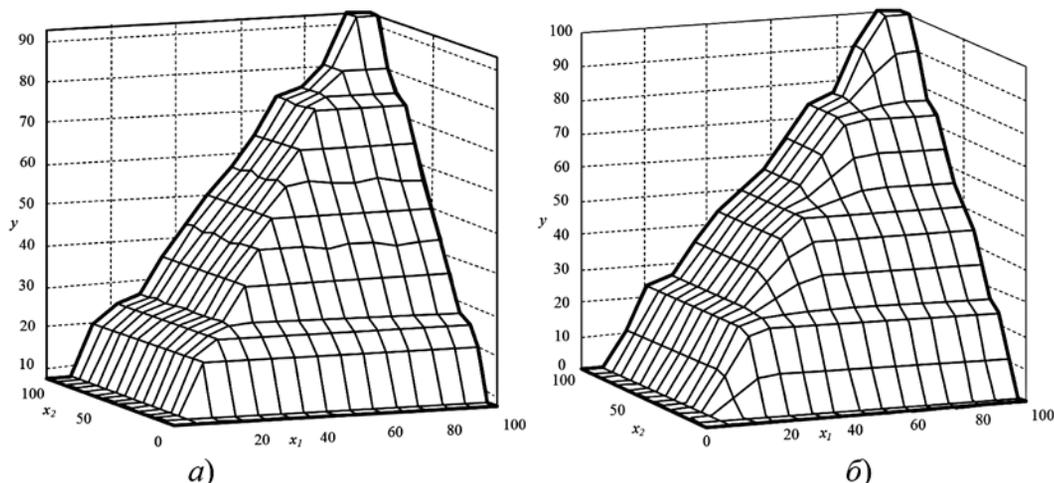


Рис. 2. Поверхности СНВ, используемых при построении иерархического вычислительного комплекса: а — СНВ на основе алгоритма Мамдани; б — СНВ на основе алгоритма Такаги—Сугено

где $NB \sim 0$, $NM1 \sim 12,25$, $NM2 \sim 25$, $NM3 \sim 37,5$, $Z \sim 50$, $PM1 \sim 62,5$, $PM2 \sim 75$, $PM3 \sim 87,5$, $PB \sim 100$ баллов.

ИСНВ состояли из пяти уровней иерархии ($\alpha = 5$). На вход первого уровня подавались 32 переменных: $X_{ИСНВ}^{вход} = \{x_i; i = \overline{1,32}$. Элементы множества $X_{ИСНВ}^{вход}$ разделялись на пары, для их агрегирования использовалось 16 СНВ 1-го уровня иерархии, которые можно представить в виде $V_{ПСНВ}^1$. Выходные значения СНВ из множества $V_{ПСНВ}^1$ и последующих уровней иерархии разделялись на пары, для агрегирования которых использовались СНВ 2-го, 3-го, 4-го и 5-го уровней иерархии.

Для формирования закономерностей на вход ИСНВ подавались различные комбинации значений переменных. Каждая часть обеих серий эксперимента включала 21 цикл моделирования. В первом цикле на все входы подавали значения в диапазоне от 0 до 100 баллов с шагом 1 балл. В остальных циклах на вход переменной x_1 подавали значения в диапазоне от 0 до 100 баллов с шагом 5 единиц, а на входы переменных $x_2 \dots x_{32}$ подавались фикси-

рованные значения. Таким образом, во втором цикле на вход переменной x_1 подавались значения в диапазоне от 0 до 100 баллов с шагом 5 единиц, а на входы переменных $x_2 \dots x_{32}$ подавались только значения 0 баллов. В третьем цикле на вход переменной x_1 снова подавались значения в диапазоне от 0 до 100 баллов с шагом 5 единиц, а на входы переменных $x_2 \dots x_{32}$ подавались только значения 5 баллов и т. д. до завершающего 21 цикла. На рис. 3 (см. вторую сторону обложки) показаны результаты 1-го цикла моделирования для ИСНВ-М: на рис. 3, *а* для описания переменных используются пять термов, на рис. 3, *б* — девять термов. Результаты 1-го цикла моделирования для ИСНВ-ТС, в которых переменные описываются пятью терминами, показаны на рис. 4: на рис. 4, *а* для описания переменных используются пять термов, на рис. 4, *б* — 9 термов. Результаты моделирования со 2-го по 21-й циклы для ИСНВ-М, использующих 5 термов и 9 термов для ИСНВ-М, приведены на рис. 5, *а* и 5, *б*, соответственно. Аналогичные результаты для ИСНВ-ТС приведены на рис. 6, *а* и 6, *б*.

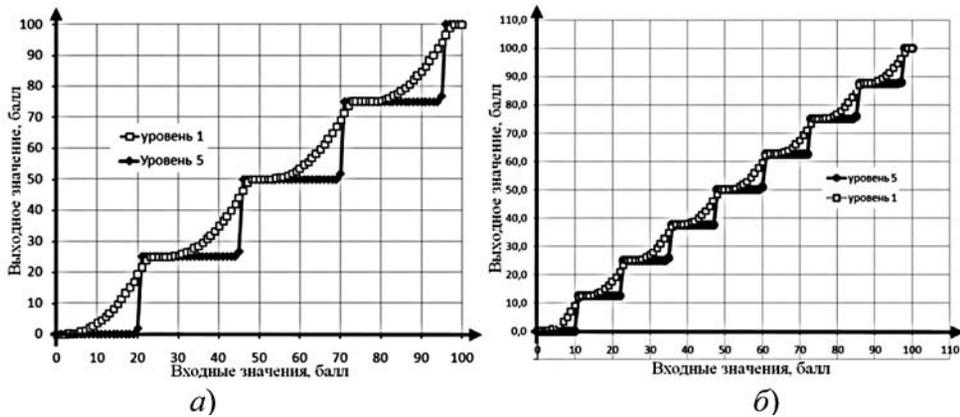


Рис. 4. Результат 1-го цикла моделирования для ИСНВ-ТС

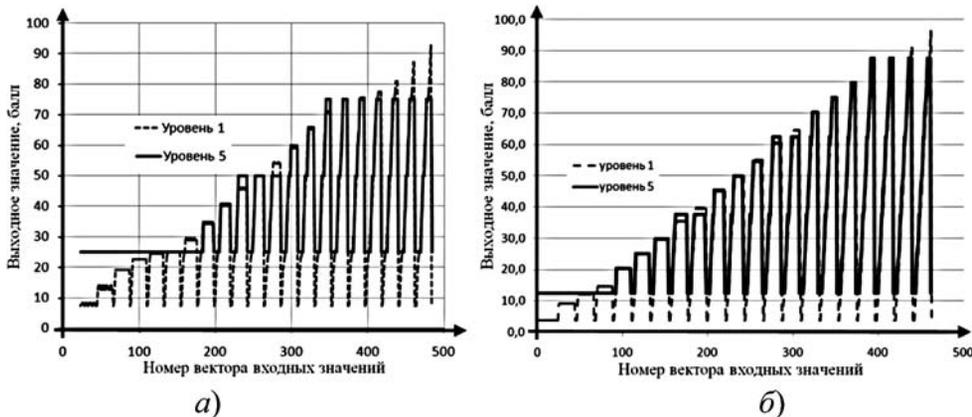


Рис. 5. Результат со 2-го по 21-й циклы моделирования для ИСНВ-М

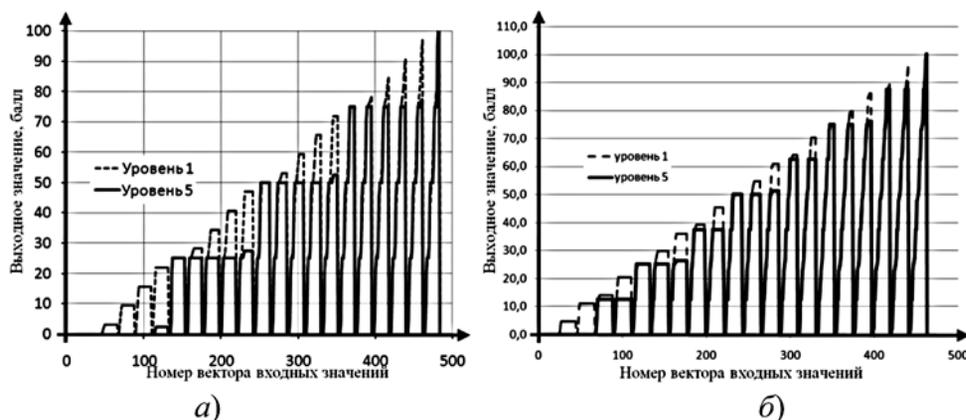


Рис. 6. Результат со 2-го по 21-й циклы моделирования для ИСНВ-ТС

Анализ рис. 3 (см. вторую сторону обложки) и рис. 4—6 выявляет следующие закономерности:

- у ИСНВ-М наблюдается сжатие диапазона выходных значений в области определения ФП термов NB и PB при увеличении числа иерархий, наблюдается закономерность, связанная с группировкой значений выходных переменных в области ядер ФП, "промежуточных" между термами NB и PB ; при увеличении уровней иерархии наблюдается завышение оценки с левой стороны от ядер "промежуточных" термов и занижение оценки с правой стороны ядер "промежуточных" термов, подобное явление хорошо заметно на рис. 3, а в области значения 50 баллов (терм Z);
- у всех ИСНВ-ТС наблюдается группировка значений в области ядер ФП, число групп зависит от числа термов, с помощью которых описывается состояние переменной, диапазон выходных значений СНВ сохраняется в районе ядер термов старше терма NB , аналогично с левой стороны наблюдается завышение оценки (менее значительное, чем у ИСНВ-М), а с правой стороны наблюдается занижение оценки (более значительное, чем у ИСНВ-М).

Проявление закономерностей потребовало их дальнейшего изучения и предложения рекомендаций по их компенсации.

Исследование причин возникновения закономерностей в ИСНВ, основанных на СНВ, использующих алгоритм Мамдани

Анализ работ [10, 11, 25] и процедур, которые выполняются в ходе функционирования алгоритма Мамдани, показал, что причиной проявления сужения диапазона выходных значений является особенность выполнения

операции дефаззификации. Эта операция основана на использовании метода определения выходного значения СНВ как центра тяжести фигуры итоговой ФП. При активации правил, выходными значениями которых являлись термы NB или PB , численные выходные значения СНВ определялись как центры тяжести фигур, полученных с использованием соответствующих ФП, показанных на рис. 1, а и рис. 1, б, описывающих эти термы. В результате значение выходной переменной не могло достигнуть максимального или минимального значений, предусмотренных ее шкалой. Например, при подаче на входы $x_1 \dots x_{32}$ ИСНВ-М, в которой для описания входных переменных используются ФП, показанные на рис. 1, а, значений 2 балла у СНВ первого уровня иерархии активируется правило

$$p_{n \min} = (x_1^{fz} = NB) \wedge (x_2^{fz} = NB) \Rightarrow (y^{fz} = NB), \quad (6)$$

и выходные значения этих СНВ ($y_1^1 \dots y_{16}^1$) становятся равными 7,29 баллов. С учетом параметров ФП термов на рис. 1, а у СНВ второго уровня иерархии кроме правила (6) активируются правила с термами NM , в том числе и правило

$$(x_1^{fz} = NM) \wedge (x_2^{fz} = NM) \Rightarrow (y^{fz} = NM).$$

Это происходит потому, что в рассматриваемом примере носитель ФП терма NM имеет значение $H_{NM} = [2,5; 47,5]$, следовательно, значение ФП $\mu_{NM}(7,29) > 0$. Вместе с этим, поскольку ядро ФП терма NB имеет значение $Y_{NB} = [0; 2,5]$, то $\mu_{NB}(7,29) < 1$. В результате активации этих правил выходные значения СНВ второго уровня иерархии $y_1^2 \dots y_8^2$ будут равны 16,2 балла. По аналогичной причине выходные значения СНВ третьего уровня иерархии $y_1^3 \dots y_4^3$ примут значения 23,2 балла.

Таким образом, результат активации правила $(x_1^{fz} = NM) \wedge (x_2^{fz} = NM) \Rightarrow (y^{fz} = NM)$ начинает сильнее влиять на выходное значение СНВ, чем результат активации правила $(x_1^{fz} = NB) \wedge (x_2^{fz} = NB) \Rightarrow (y^{fz} = NB)$. Учитывая, что носитель ФП терма NB равен $H_{NB} = [0; 22,5]$, а значение ядра ФП терма NM равно $Y_{NM} = [22,5; 27,5]$, то у СНВ 4-го уровня иерархии правило (6) активировано не будет, а активируются только правила с термами NM . В результате выходное значение СНВ 4-го уровня иерархии $y_1^4 \dots y_2^4$ принимает значение 25 баллов. При подаче этого значения на СНВ 5-го уровня иерархии выходное значение уже не изменяется и остается равным 25 баллов, так как $\mu_{NM}(25) = 1$, а значения ФП других термов для 25 баллов равны 0. Аналогичный процесс происходит при подаче на входы $x_1^1 \dots x_{32}^1$ значения 98 баллов, при этом смещение наблюдается в сторону ядра терма PM .

Для уменьшения проявления сжатия диапазона выходных значений применительно к ИСНВ-М предложена следующая рекомендация. Пусть $y_{out\ n_{min}}^{c/CHB_{\alpha-1}}$ и $y_{out\ n_{max}}^{c/CHB_{\alpha-1}}$ являются выходными значениями СНВ предыдущего уровня иерархии, полученными в результате активации правил, выходными значениями которых являются термы с минимальным t_{min} или максимальным t_{max} семантическим значением. Тогда в СНВ следующего уровня иерархии эти значения должны принадлежать только ядрам ФП аналогичных термов. Значение ФП других термов для значений $y_{out\ n_{min}}^{c/CHB_{\alpha}}$ и $y_{out\ n_{max}}^{c/CHB_{\alpha}}$ должны равняться нулю:

$$\begin{aligned} & [(y_{out\ n_{min}}^{c/CHB_{\alpha-1}} \in Y_{t_{min}}^{CHB_{\alpha}}) \wedge (y_{out\ n_{max}}^{c/CHB_{\alpha-1}} \in Y_{t_{max}}^{CHB_{\alpha}})] \wedge \\ & \wedge [(\mu_{t_k}(y_{out\ n_{min}}^{c/CHB_{\alpha-1}}) = 0 | t_k \neq t_{min}) \wedge \\ & \wedge (\mu_{t_k}(y_{out\ n_{max}}^{c/CHB_{\alpha-1}}) = 0 | t_k \neq t_{max})] = 1. \end{aligned} \quad (7)$$

Для проверки (7) использована ИСНВ-М, построенная на основе СНВ, использующих по пять термов для описания переменных. В этой ИСНВ-М проведена корректировка носителей ФП термов NM и PM в СНВ всех уровней иерархии. Для соблюдения условия разбиения единицы скорректированы и ядра ФП термов NB и PB . Дополнительно для проверки предположения о завышении оценок в "левой" и "правой" областях термов, которые не граничат с крайними термами для терма Z , скорректированы параметры ядра. После корректировки были получены следующие параметры ФП: для терма NB — ядро $Y_{NB} = [0; 8]$, носитель $H_{NB} = [0; 22,5]$; для терма NM — ядро $Y_{NM} =$

$[22,5; 27,5]$, носитель $H_{NM} = [8; 45]$; для терма Z — ядро $Y_Z = [45; 55]$, носитель $H_Z = [27,5; 72,5]$; для терма PM — ядро $Y_{PM} = [72,5; 77,5]$, носитель $H_{PM} = [55; 91]$; для терма PB — ядро $Y_{PB} = [72,5; 100]$, носитель $H_{PB} = [91; 100]$.

Вид ФП термов входных переменных после корректировки показан на рис. 7, а (см. вторую сторону обложки), результат моделирования ИСНВ, когда на вход подавались значения $x_1 \dots x_{32}$ от 0 до 100 баллов с шагом 1 балл, для перехода $NB \rightarrow NM$ показан на рис. 7, б, для перехода $NM \rightarrow Z$ — на рис. 7, в, для перехода $Z \rightarrow PM$ — на рис. 7, г, а для перехода $PM \rightarrow PB$ — на рис. 7, д. Анализ рис. 7 (см. вторую сторону обложки) подтверждает результативность принятых мер по корректировке ФП термов NB , NM , PM и PB по снижению сжатия диапазона выходных значений после прохождения пяти уровней иерархии.

Также анализ рис. 7 показывает, что увеличение ядра ФП терма Z при увеличении уровня иерархии привело к тому, что с левой стороны от терма Z наблюдается завышение, а с правой стороны — занижение значения выходной оценки. Объясняется это тем, что изменение размеров ядра изменяет значение степени принадлежности входного численного значения к терму Z , что с учетом (7) влияет на уровень активации правил, содержащих этот терм. Это оказывает влияние на форму ФП терма, входящего в состав итоговой фигуры, описывающей выходную переменную в нечеткой форме, что при использовании метода центра тяжести для дефаззификации влияет на значение численной оценки. Терм большего размера вносит более весомый вклад в форму фигуры и смещает в "свою" сторону значение проекции центра тяжести. После попадания на следующий уровень иерархии это значение обладает еще большей принадлежностью к ФП терма Z , что приводит к "большому" увеличению влияния правил с этими термами.

Эксперимент показал, что ИСНВ-М имеет высокую чувствительность к изменению положений ФП входных переменных. В результате требуется отладка комплекса, в основе которого они заложены. Сами положения ФП после корректировки являются искусственными, так как вносимые изменения — это своеобразная "подгонка" под определенный результат, следовательно, после корректировки ФП термы не в полной мере отражают мнения экспертов, определяющих степень принадлежности определенного численного значения переменной к соответствующим термам.

**Исследование закономерностей в ИСНВ,
основанных на СНВ,
использующих алгоритм Такаги—Сугено**

Анализ рис. 4 и рис. 6 позволяет предварительно заключить, что ИСНВ-ТС обладают более высокой "стабильностью", число групп выходных переменных связано с числом термов, используемых для описания переменных. Для исследования этой закономерности предлагается проведение моделирования в аналогичных условиях, что и для ИСНВ-М. Результат моделирования ИСНВ-ТС при использовании для описания входных переменных ФП термов, показанных на рис. 7, а (см. вторую сторону обложки), и подаче на вход значений x_1, \dots, x_{32} от 0 до 100 баллов с шагом 1 балл для переходов $NB \rightarrow NM, NM \rightarrow Z, Z \rightarrow PM$ и $PM \rightarrow PB$ показаны на рис. 8, а—г (см. третью сторону обложки). Критерием сравнения результатов моделирования ИСНВ-ТС и ИСНВ-М выступила ширина диапазона значений входных переменных, при котором не изменяются значения выходной переменной в окрестности термов NB, NM, Z, PM, PM , результаты представлены в таблице.

Как видно из таблицы и рис. 8 (см. третью сторону обложки), у ИСНВ-ТС относительно ИСНВ-М увеличение размеров ядер термов NB и PB не повлияло на изменение диапазона значений выходной переменной. Также корректировка ФП терма NB в ИСНВ-ТС не оказывает

значительного влияния на область, в которой наблюдается занижение оценки при увеличении уровня иерархии, при этом корректировка ФП терма PB привела к тому, что завышение оценки начинает наблюдаться при подаче на вход оценок со значением от 87 баллов. Изменение размеров ядра ФП терма Z приводит к увеличению области завышения оценки с левой стороны и практически не влияет на увеличение области занижения оценки с правой стороны.

Следует отметить, что ИСНВ-М обеспечивают плавный переход между областями группировки выходных оценок, а у ИСНВ-ТС во всех экспериментах отмечен ступенчатый характер изменения оценки. У ИСНВ-М тенденция к стремлению получить ступенчатый характер выходной закономерности происходит заметно медленнее, чем у ИСНВ-ТС.

Обобщение результатов исследований

Для обобщения результатов исследований поставлен дополнительный эксперимент. В рамках эксперимента исследовали ИСНВ-М и ИСНВ-ТС, аналогичные тем, которые использовались в предыдущих экспериментах. Для описания входных переменных применяли ФП, вид которых показан на рис. 9, а (см. третью сторону обложки). В качестве входных значений были использованы случайные числа из диапазона от 0 до 100 баллов. Результат

Результаты сравнения ИСНВ-М и ИСНВ-ТС

Параметры ФП термов входных переменных		Ширина диапазона фиксации выходного значения	
Носитель ФП	Ядро ФП	ИСНВ-М	ИСНВ-ТС
До корректировки ФП			
$H_{NB} = [0; 22,5]$	$Я_{NB} = [0; 2,5]$	$y < 25; x \in \emptyset$	$y = 0; x \in [0; 30]$
$H_{NM} = [2,5; 47,5]$	$Я_{NM} = [22,5; 27,5]$	$y = 25; x \in [0; 31]$	$y = 25; x \in [21; 44]$
$H_Z = [27,5; 72,5]$	$Я_Z = [47,5; 52,5]$	$y = 50; x \in [44; 56]$	$y = 50; x \in [46; 69]$
$H_{PM} = [52,5; 97,5]$	$Я_{PM} = [72,5; 77,5]$	$y = 75; x \in [69; 100]$	$y = 75; x \in [71; 94]$
$H_{PB} = [77,5; 100]$	$Я_{PB} = [97,5; 100]$	$y > 75; x \in \emptyset$	$y = 75; x \in [96; 100]$
После корректировки ФП			
$H_{NB} = [0; 22,5]$	$Я_{NB} = [0; 8]$	$y < 7,3; x \in \emptyset$	—
		$y = 7,3; x \in [0; 8]$	$y = 0; x \in [0; 21]$
$H_{NM} = [8; 45]$	$Я_{NM} = [22,5; 27,5]$	$y = 25; x \in [9; 30]$	$y = 25; x \in [22; 41]$
$H_Z = [27,5; 72,5]$	$Я_Z = [45; 55]$	$y = 50; x \in [37; 63]$	$y = 50; x \in [42; 70]$
$H_{PM} = [55; 91]$	$Я_{PM} = [72,5; 77,5]$	$y = 75; x \in [70; 90]$	$y = 75; x \in [72; 86]$
$H_{PB} = [77,5; 100]$	$Я_{PB} = [91; 100]$	$y = 92,7; x \in [91; 100]$	$y = 100; x \in [87; 100]$
		$y > 92,7; x \in \emptyset$	—

моделирования ИСНВ-ТС показан на рис. 9, б (график "уровень 5"), а результат моделирования ИСНВ-М показан на рис. 9, в (график "уровень 5").

Как видно на рис. 9, б, в, наблюдаются области группировки оценок, но в переходе между областями $NM \rightarrow Z$ и частично $Z \rightarrow PM$ наблюдается многообразие значений. Таким образом, исследуемые ИСНВ не справились с задачей отнесения объекта к определенной группе состояния. С учетом анализа результатов проделанных ранее экспериментов было принято решение о добавлении к уже созданным вычислительным системам группы последовательно соединенных СНВ, аналогичных тем, что уже входили в состав ИСНВ. Таким образом, множество промежуточных вершин в ИСНВ предлагается представить в виде

$$V_{\text{ПСНВ}} = V_{\text{ПСНВ}}^{\text{Агр}} \cup V_{\text{ПСНВ}}^{\text{Доп}}, \quad (8)$$

где $V_{\text{ПСНВ}}^{\text{Агр}}$ — СНВ, которые используются для агрегирования значений входных переменных, а $V_{\text{ПСНВ}}^{\text{Доп}}$ — дополнительные СНВ, которые определяют отнесение состояния объектов к одной из групп. После этого проведен повтор эксперимента и выявлено, что группировка значений у ИСНВ-ТС наблюдается после добавления к выходу подмножества из пяти последовательно соединенных СНВ-ТС. Результат моделирования показан на рис. 9, б (график "уровень 5×5 "), цифра 5 означает число дополнительных СНВ, добавленных к выходу ИСНВ-ТС. Как видно из рис. 9 (см. третью сторону обложки), после добавления дополнительной цепочки СНВ к ИСНВ-ТС удалось добиться распределения выходных значений по определенным группам состояния. Аналогичное добавление к ИСНВ-М не позволило добиться подобной группировки даже после добавления последовательности из семи СНВ-М, результат приведен на рис. 9, в (график "уровень 5×7 ").

Обсуждение

В рамках проведения экспериментов накладывались ограничения (i)—(iii). Снятие ограничения (i), связанного с использованием одного диапазона значений входных переменных, для более общего случая может быть реализовано с помощью использования методов нормирования значений входных переменных к единой шкале, описание подобных методов приведено в работах [10, 26, 27]. Несущественность ограни-

чения (ii), связанного с наличием положительной корреляции между входными и выходными переменными, и ограничения (iii), связанного с использованием в составе множества $V_{\text{ПСНВ}}$ однотипных СНВ, объясняется тем, что "ступенчатый" характер выходная закономерность приобретает за счет специфики формы ФП термов агрегируемых переменных и видов операций, которые обобщают значения степеней принадлежности значений входных переменных при определении значения выходной переменной.

Областью использования предлагаемых положений является ранжирование анализируемых объектов по группам состояний в условиях, когда для формирования оператора агрегирования входных переменных используются знания экспертов. Полученные результаты могут быть использованы, например, при реализации систем поддержки принятия решений по оценке состояния элементов территориально распределенных систем. В зависимости от отнесения к одной из групп возможно определение очередности и типа управляющих воздействий.

Заключение

Проведен анализ закономерностей при формировании выходных значений ИСНВ. В ходе экспериментов установлено, что ИСНВ обладают свойством группировки выходных значений. Сравнению подвергались иерархические вычислительные комплексы, сформированные с использованием СНВ на основе алгоритмов Такаги—Сугено и Мамдани. Свойство группировки выходных значений более выражено у ИСНВ, основанных на СНВ, использующих алгоритм Такаги—Сугено. Подобные ИСНВ более склонны к занижению выходной оценки практически на всей шкале выходной переменной, кроме перехода между термом, имеющим максимальное семантическое значение, и соседним с ним термом. ИСНВ, в основе которых заложены СНВ, использующие алгоритм Мамдани, формируют большее разнообразие выходных значений, но при этом демонстрируют сильную зависимость от расположения термов ФП входных переменных. При отдельных условиях подобные ИСНВ демонстрировали склонность к сжатию диапазона значений выходных переменных. В рамках работы показаны условия, которые необходимо соблюдать при формировании ИСНВ, чтобы избежать подобного явления.

Областью использования результатов является формирование систем поддержки приня-

тия решений по оценке целесообразности оказания управляющих воздействий на контролируемые элементы объектов инфраструктуры проектов различного назначения.

Список литературы

1. **Тенев В. А., Шаталова О. М.** Методы нечеткого логического вывода при построении экспертных систем прогнозирования инновационных процессов // Интеллектуальные системы в производстве. 2019. Т. 17, № 4. С. 129–136.
2. **Сизов А. С., Добрица В. П., Добросердов О. Г., Агакищев О. И., Зернушкин А. Е., Халин Ю. А., Сильченко Р. С.** Нечеткая система оценки и управления привлекательностью технических или экономических объектов (на примере выбора фирмы-поставщика ресурсов) // Известия Юго-Западного государственного университета. 2019. Т. 23, № 1. С. 95–106.
3. **Благодатский Г. А., Горохов М. М., Переведенцев Д. А.** Моделирование системы нечеткого логического вывода оценки наукоемких проектов // Автоматизация процессов управления. 2017. № 2 (48). С. 82–89.
4. **Михелев В. В., Синюк В. Г.** Методы вывода для систем логического типа на основе нечеткой степени истинности // Известия Российской академии наук. Теория и системы управления. 2018. № 3. С. 108–115.
5. **Хрипков Д. В.** Формирование функций принадлежности в системах диагностики на базе нечеткого логического вывода // Вестник Московского энергетического института. Вестник МЭИ. 2017. № 4. С. 110–116.
6. **Kosko B.** Fuzzy systems as universal approximators // IEEE Transactions on Computers. 1994. Vol. 43, Iss. 11. P. 1329–1333.
7. **Ying H.** General Takagi-Sugeno fuzzy systems with simplified linear rule consequent are universal controllers, models and filters // Information Sciences. 1998. Vol. 108, Iss. 1–4. P. 91–107.
8. **Ying H.** General MISO Takagi-Sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modeling applications // IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. 1997. P. 1335–1340.
9. **Шведов А. С.** Аппроксимация функций с помощью нейронных сетей и нечетких систем // Проблемы управления. 2018. № 1. С. 21–29.
10. **Пегат А.** Нечеткое моделирование и управление. М.: Бинум. Лаборатория знаний, 2017. 800 с.
11. **Штовба С. Д.** Проектирование нечетких систем средствами MATLAB. М.: Горячая линия-Телеком, 2007. 288 с.
12. **Бобырь М. В., Кулабухов С. А., Милостная Н. А.** Обучение нейро-нечеткой системы на основе метода разности площадей // Искусственный интеллект и принятие решений. 2016. № 4. С. 15–26.
13. **Li W., Qiao J., Zeng X.-J., Du S.** Identification and simplification of T-S fuzzy neural networks based on incremental structure learning and similarity analysis // Fuzzy Sets and Systems. 2020. P. 65–86.
14. **Ходашинский И. А., Бардамова М. Б., Ковалев В. С.** Отбор признаков и построение нечеткого классификатора на основе алгоритма прыгающих лягушек // Искусственный интеллект и принятие решений. 2018. № 1. С. 79–83.
15. **Галушка В. В., Фатхи В. А.** Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных // Инженерный вестник Дона. 2013. № 2 (25). [Электронный ресурс]: http://www.ivdon.ru/uploads/article/pdf/IVD_7_Galushka.pdf_1597.pdf (дата обращения: 25.12.2020).
16. **Меньшутина Н. В., Гордиенко М. Г., Иванов С. И., Матасов А. В.** Анализ и обработка данных для прогнозирования состояния больших // Программные продукты и системы. 2016. № 1. С. 180–185.
17. **Михайлов А. С., Староверов Б. А.** Визуализация процесса формирования обучающей выборки для искусственной нейронной сети // Научная визуализация. 2016. Т. 8, № 2. С. 85–97.
18. **Дагаева М. В., Катасева Д. В., Катасев А. С., Кирпичников А. П.** Нечеткая экспертная система диагностики маслонаполненных силовых трансформаторов // Вестник Технологического университета. 2018. Т. 21, № 2. С. 148–154.
19. **Димитров В. П., Борисова Л. В., Нурутдинова И. Н.** Формирование базы знаний интеллектуальной системы для технологической регуляторки зерноуборочного комбайна // Инновации в АПК: проблемы и перспективы. 2018. № 3 (19). С. 3–14.
20. **Жигалов И. Е., Моногаров О. И.** Экспертная система диагностики причин образования опасных микродефектов в кабельной изоляции // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2020. Т. 20, № 1. С. 51–59.
21. **Wyrol B.** Implementation of the FATI hierarchical fuzzy inference system using the immutability decomposition method // Fuzzy Sets Syst. 2020. Vol.381. P. 105–123.
22. **Alrashoud M.** Hierarchical Fuzzy Inference System for Diagnosing Dengue Disease // IEEE International Conference on Multimedia & Expo Workshops (ICMEW). 2019. P. 31–36.
23. **Mutlu B., Sezer E. A., Akcayol M. A.** End-to-End Hierarchical Fuzzy Inference Solution // IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). 2018.
24. **Amindousta A., Ahmeda S., Saghafiniab A., Bahreininejada A.** Sustainable supplier selection: A ranking model based on fuzzy inference system // Applied Soft Computing. 2012. N. 12. P. 1668-1677.
25. **Леоненков А. В.** Нечеткое моделирование в среде MATLAB и fuzzyTech. СПб.: БХВ-Петербург, 2005. 736 с.
26. **Lee K. H.** First Course on Fuzzy Theory and Applications. Springer Science & Business Media, 2004. 335 p.
27. **Сорокин А. А.** Нормализация входных переменных для систем нечеткого вывода на основе кусочных функций // Математические методы в технике и технологиях — ММТТ. 2020. Т. 12-2. С. 44–47.

A. A. Sorokin, Associate Professor, e-mail: alsorokin.astu@mail.ru,
Astrakhan State Technical University, Astrakhan, Russian Federation

Investigation of Hierarchical Fuzzy Inference Systems, when Obtaining Integral Estimates of the Analyzed Objects

The purpose of this paper is to study the patterns of the formation of output values in hierarchical systems of fuzzy inference. Hierarchical fuzzy inference systems (HFIS) are used to aggregate heterogeneous parameters during the assessment of the state of various elements of complex systems. The use of HFIS allows avoiding the "curse" of the dimension associated with a

strong increase in the number and complication of the structure of the production rule, which is characteristic of conventional fuzzy inference systems (FIS), which aggregate the results of interaction of different values of input variables in one knowledge base. As part of the research, numerical experiments were carried out to study the features of the formation of output patterns in HFIS, based on FIS using the Mamdani and Takagi-Sugeno algorithms. As a result of the experiment, it was shown that the output values of the studied HFIS tend to be grouped in the region of fixed values, and the output pattern itself acquires a stepwise character. The revealed property allows using HFIS to distribute the objects of the analyzed sample into groups of states. This property can be used to solve problems of distributing objects into groups in conditions when it is difficult to form a training sample for machine learning methods, but at the same time there is knowledge of the expert group about the features of the functioning of the object of research. Additionally, the paper investigates the features of the formation of output patterns depending on the parameters of the membership functions describing the input variables in HFIS, which are based on FIS using the Mamdani algorithm and HFIS, which are based on FIS using the Takagi-Sugeno algorithm.

Keywords: information processing, hierarchical fuzzy inference system, Mamdani algorithm, Takagi-Sugeno algorithm, modeling, membership function, group distribution, expert group

DOI: 10.17587/it.27.582-591

References

1. Tenenev V. A., Shatalova O. M. Fuzzy inference methods for forming expert systems for forecasting innovative processes, *Intelligent Systems in Manufacterin*, 2019, vol. 17, no. 4, pp. 129–136 (in Russian).
2. Sizov A. S., Dobritsa V. P., Dobroserdov O. G., Atakishchev O. I., Zernushkin A. E., Khalin Yu. A., Sylchenko R. S. Fuzzy System for Assessment and Management of Attractiveness of Technical or Economic Objects (Case Study of the Selection of a Resource Supplier), *Proceedings of the Southwest State University*, 2019, vol. 23(1), no. 2, pp. 95–106 (in Russian).
3. Blagodatskii G. A., Gorokhov M. M., Perevedentsev D. A. Modelling the system of fuzzy logical inference for evaluating science-intensive projects, *Automation of control processes*, 2017, no. 2, pp. 82–89 (in Russian).
4. Mikhelev V. V., Sinyuk V. G. Determination methods for logical-type systems based on fuzzy degree of truth, *Journal of Computer and Systems Sciences International*, 2018, no. 3, pp. 108–115 (in Russian).
5. Khripkov D. V. Shaping the Membership Functions in Fuzzy Logic Inference Based Diagnostic Systems, *MPEI Vestnik*, 2017, no. 4, pp. 110–116 (in Russian).
6. Kosko B. Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, 1994, vol. 43, iss. 11, pp. 1329–1333.
7. Ying H. General Takagi-Sugeno fuzzy systems with simplified linear rule consequent are universal controllers, models and filters, *Information Sciences*, 1998, vol. 108, iss. 1–4, pp. 91–107.
8. Ying H. General MISO Takagi-Sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modeling applications, *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 1997, pp. 1335–1340.
9. Shvedov A. S. Functions approximating by neural networks and fuzzy systems, *Control sciences*, 2018, no. 1, pp. 21–29 (in Russian).
10. Pegat A. Fuzzy Modeling and Control, Moscow, Binom. Knowledge lab, 2017, 800 p. (in Russian).
11. Shtovba S. D. Design of fuzzy systems by means of MATLAB, Moscow, Goryachayali-niya-Telekom Publ, 2007, 288 p. (In Russian).
12. Bobyr M. V., Kulabuchov S. A., Milostnaya N. A. Teaching of neuro-fuzzy system on the basis of the method of difference areas. *Artificial Intelligence and Decision Making*, 2016, no. 4, pp. 15–26 (in Russian).
13. Li W., Qiao J., Zeng X.-J., Du S. Identification and simplification of T-S fuzzy neural networks based on incremental structure learning and similarity analysis, *Fuzzy Sets and Systems*, 2020, vol. 394, pp. 65–86.
14. Hodashinsky I. A., Bardamova M. B., Kovalev I. V. Using shuffled frog-Leaping algorithm for feature selection and fuzzy classifier design, *Artificial Intelligence and Decision Making*, 2018, no. 1, pp. 79–83.
15. Galushka V. V., Fathi V. A. Formation of a training sample when using artificial neural networks in problems of finding database errors, *Don's Engineering Bulletin (Inzhenernyj vestnik Dona)*, 2013, no. 2 (25), Available at: http://www.ivdon.ru/uploads/article/pdf/IVD_7_Galushka.pdf_1597.pdf (in Russian).
16. Menshutina N. V., Gordienko M. G., Ivanov S. I., Matasov A. V. Data analysis and processing to predict patients' health status, *Software & Systems*, 2016, no. 1, pp. 180–185 (in Russian).
17. Mikhailov A. S., Staroverov B. A. Visualization of training sample creation process for artificial neural network, *Scientific Visualization*, 2016, vol. 8, no. 2, pp. 85–97 (in Russian).
18. Dagaeva M. V., Kataseva D. V., Kataseva A. S., Kirpichnikov A. P. Fuzzy expert system of oil-filled power transformer diagnostics. *Bulletin of the Technological University*, 2018, vol.21, no. 2, pp. 148–154 (in Russian).
19. Dimitrov V. P., Borisova L. V., Nurutdinova I. N. Knowledge base development of intelligent system for technological adjustment of the grain harvester, *Innovations in Agricultural Complex: problems and perspectives*, 2018, no. 3, pp. 3–14 (in Russian).
20. Zhigalov I. E., Monogarov O. I. The expert diagnostic system of the reasons for dangerous microdefects in cable insulation, *Bulletin of the South Ural State University. Series Computer Technology, Automatic Control, Radio Electronics*, 2020, vol. 20, no. 1, pp. 51–59 (in Russian).
21. Wyrwol B. Implementation of the FATI hierarchical fuzzy inference system using the immutability decomposition method, *Fuzzy Sets Syst.*, 2020, vol. 381, pp. 105–123.
22. Alrashoud M. Hierarchical Fuzzy Inference System for Diagnosing Dengue Disease, *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2019, pp. 31–36.
23. Mutlu B., Sezer E. A., Akcayol M. A. End-to-End Hierarchical Fuzzy Inference Solution, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018.
24. Amindousta A., Ahmeda S., Saghafiniab A., Bahreininejada A. Sustainable supplier selection: A ranking model based on fuzzy inference system, *Applied Soft Computing*, 2012, no. 12, pp. 1668–1677.
25. Leonenkov A. V. Fuzzy modeling in MATLAB and fuzzy-Tech, SPb., BHV-Petersburg, 2005, 736 p. (in Russian).
26. Lee K. H. First Course on Fuzzy Theory and Applications, Springer Science & Business Media, 2004, 335 p.
27. Sorokin A. A. Normalizing input variables for fuzzy output systems using piece functions, *Mathematical methods in engineering and technology — MMET*, 2020, vol. 12-2, pp. 44–47 (in Russian).

Р. Э. Асратян, канд. техн. наук, вед. науч. сотр., e-mail: rubezas@yandex.ru,
Институт проблем управления им. В. А. Трапезникова РАН

Сетевая служба параллельной обработки защищенных информационных запросов в мультисерверной среде

Рассмотрены принципы организации интернет-службы, предназначенной для реализации параллельной обработки защищенных запросов в распределенных информационных системах, ориентированных на работу в сложных сетевых средах со многими обрабатываемыми серверами. Описано применение этой службы для комбинирования последовательной ("конвейерной") и параллельной обработки запросов в мультисерверной среде.

Ключевые слова: распределенные системы, сетевые технологии, Web-технологии, параллельные вычисления, сетевое взаимодействие, информационная безопасность

Введение

Современные сетевые технологии (например, технология Web-сервисов в рамках сетевой архитектуры .Net) предоставляют широкий набор методов и средств для поддержки сетевого взаимодействия [1, 2]. Тем не менее, разработчики информационных систем нередко испытывают трудности с организацией защиты и распределенной обработки данных в сети. Эти трудности чаще всего бывают связаны с отсутствием в используемых технологиях встроенных средств защиты и аутентификации сетевых сообщений, а также простых средств организации последовательной и/или параллельной обработки информационных запросов во множестве сетевых узлов. Более всего эти трудности проявляются при создании систем, предназначенных для работы в сложных мультисерверных и мультисетевых средах в условиях высоких требований к информационной безопасности [3–6].

В работе [7] описана новая сетевая служба PMS (Protected Message Service), разработанная в целях преодоления вышеуказанного недостатка. Суть подхода заключается в тесной интеграции функций сетевого информационного обмена с функциями защиты и аутентификации данных. Внешне эта интеграция проявляется в том, что отмеченные функции входят в набор методов главного программного класса службы — класса "Защищенное сообщение", отображающего электронный документ (информаци-

онный запрос или ответ), снабженный одной или несколькими удостоверяющими электронными цифровыми подписями (ЭЦП). В отличие, например, от технологии Web-сервисов описываемая служба опирается не на модель вызова функций удаленных объектов (Remote Procedure Call), а на модель обмена сообщениями. В данном случае это означает, что все сервисные обрабатываемые функции (методы) имеют одинаковую, жесткую спецификацию: они получают объект класса "Защищенное сообщение" в качестве параметра и возвращают объект того же класса в качестве результата. Эти обрабатываемые функции группируются в одну или несколько динамических библиотек, которые подключаются к серверу PMS в момент его запуска (каждая библиотека может рассматриваться как отдаленный аналог Web-сервиса в .NET), и становятся доступными для клиентских компонентов.

Базисное свойство PMS — фиксированная спецификация сервисных функций с объектами одного и того же класса на входе и на выходе функции — дает принципиальную возможность организации "программного конвейера", основанного на направлении объекта класса "Защищенное сообщение" с выхода одной функции на вход другой. Другими словами, речь идет об обработке информационного запроса не одной сервисной функцией, но цепочкой функций таким образом, что защищенное сообщение, возвращенное каждой сервисной

функцией, или передается непосредственно на вход следующей функции в цепочке, если она имеется, или возвращается клиенту (PMS-конвейер). Разумеется, здесь имеется прямая аналогия с известным еще с первых версий системы UNIX механизмом "трубопровода" (pipeline), основанном на последовательном соединении стандартных выводов и вводов у нескольких процессов в компьютере.

Логика "конвейера" всегда предполагала строго последовательную обработку данных. Однако возможности сложных сетевых сред с десятками и сотнями сетевых узлов, в которых работают многие современные распределенные системы, пробуждают интерес к средствам параллельной обработки как к источнику повышения производительности [8, 9]. В данной работе рассматриваются основы организации новой версии PMS, включающей механизмы распределенной параллельной обработки защищенных информационных запросов в мультисерверной среде.

1. Последовательная конвейерная обработка запросов в PMS

Как и всякая сетевая служба, основанная на базовом сетевом протоколе TCP/IP [10], PMS поддерживается клиентским и серверным программным обеспечением. Сервер PMS представляет собой постоянно активную программу, обслуживающую запросы на обработку от клиентов (по умолчанию используется порт 8132). Клиентское программное обеспечение представляет собой библиотеку функций PmsBase.dll, реализующих прикладной программный интерфейс (API) к PMS. Этот интерфейс является "лицом" PMS с точки зрения пользователя.

На рис. 1 представлен фрагмент кода в нотации C#, иллюстрирующий последовательную конвейерную обработку информационного запроса с применением средств защиты данных. Две первые строки кода определяют две переменные типа PmsMessage, представляющего собой главный класс PMS "Защищенное сообщение". Первой переменной (Request) присваивается значение: объект класса PmsMessage, инициализированный символьной строкой (например, содержащей XML-документ информационного запроса). Вторая переменная (Reply) предназначена

для хранения результата обработки. В третьей строке определяется и инициализируется переменная MyConn класса PmsConnection, предназначенного для создания и прекращения сетевого соединения с сервером.

В четвертой и пятой строках кода выполняется формирование подписей в запросе. Сначала определяется переменная SenderCerts класса PmsCertList. Этот класс предназначен для хранения в памяти списков сертификатов с открытыми ключами в стандарте X509 и содержит несколько конструкторов для загрузки сертификатов из файлов или из системных хранилищ с поиском по имени владельца или по серийному номеру. В переменную SenderCerts загружаются два сертификата, соответствующих именам владельцев "Иванов" и "Петров" для последующего формирования двух ЭЦП в запросе. (Такой способ использования означает, что с этими сертификатами обязательно должны быть связаны парные им закрытые ключи, иначе формирование ЭЦП закончится неудачно.) Вызов метода AddSignatures позволяет сформировать две ЭЦП в сообщении Request.

Собственно, вызов обрабатывающей функции начинается с вызова метода Connect, устанавливающего сетевое соединение с сервером с указанным сетевым именем или адресом. Далее выполняется первая сетевая операция — запрос сертификата сервера (метод GetServerCertificate) с занесением результата в переменную ReceiverCert уже знакомого нам класса PmsCertList для последующего шифрования информационного запроса. Сразу же после

```
PmsMessage Request= new PmsMessage("<query> ... </query>");
PmsMessage Reply;
PmsConnection MyConn = new PmsConnection();
PmsCertList SenderCerts = PmsCertList( new string [] {"Иванов", "Петров"});
Request.AddSignatures(SenderCerts);

MyConn.Connect("Srv1");
PmsCertList ReceiverCert = MyConn.GetServerCertificate();

Reply=Request.Process(MyConn,
"Lib1.Func1, Srv2/Lib2.Func2, Srv3/(Lib3.Func3,Lib4.Func4), Lib5.Func5 param",
ReceiverCert);
if(Reply != null)
{
    string Signer;
    for(int i=0; Reply.Signatures.Length; i++)
        if(Reply.VerifySignature(i, out Signer) >= 0)
            Console.WriteLine("Ответ подписал: " + Signer);
    Console.WriteLine(Reply.GetString());
}
else
    Console.WriteLine("Ошибка: " +MyConn.ErrMsg);
MyConn.Disconnect();
```

Рис. 1. Пример последовательной конвейерной обработки в PMS

успешного получения сертификата сервера выполняется вызов удаленной функции с помощью применения метода Process к переменной Request с использованием все того же соединения и полученного сертификата.

Самым важным параметром метода Process является "конвейерная строка", задающая последовательность запуска сервисных функций для обработки запроса. Первая функция в последовательности получает в качестве фактического параметра значение клиентской переменной Request, а каждая последующая — результат выполнения предыдущей. Приведенная на рис. 1 иллюстративная последовательность включает четыре элемента, разделенных запятыми, демонстрирующих различные возможности PMS. Первый из этих элементов обозначает вызов функции Func1 из серверной библиотеки Lib1 непосредственно на обрабатываемом сервере, второй — временную передачу запроса в удаленный сервер Srv2 для его обработки там с помощью функции Func2 из библиотеки Lib2, третий элемент также обозначает временную передачу запроса в удаленный сервер (Srv3) для последовательной обработки его с помощью двух функций. Последовательность завершается вызовом функции Func5 из серверной библиотеки Lib5 непосредственно на обрабатываемом сервере с передачей ей опционального строкового параметра (param).

Результат обработки заносится в переменную Reply. Подчеркнем, что шифрование запроса и дешифрование ответа выполняются автоматически в методе Process. Последующие строки обеспечивают последовательную проверку всех ЭЦП в полученном ответе сервера (вызов метода VerifySignature) и запись в стандартный вывод сведений о подписантах.

Пример заканчивается записью результата обращения в стандартный вывод (предполагается, что результат, как и запрос, имеет форму символьной строки) и закрытием соединения с сервером с помощью метода Disconnect, так как в данном примере оно больше не нужно.

Необходимо отметить, что из фрагмента кода намеренно удалены операторы обработки исключений.

Как уже отмечалось, все сервисные обрабатываемые функции группируются в одну или несколько динамических библиотек, которые подключаются к серверу PMS в момент его запуска. Никакие специальные конструкции типа Web-сервисов не используются. Поиск и загрузка всех динамических библиотек выполняются при запуске сервера PMS из каталога,

указанного в конфигурации сервера. В каждой найденной библиотеке проводятся поиск и подключение всех функций, имеющих строго определенную спецификацию:

PmsMessage *имя_функции* (PmsMessage Inpt, string [] Conf, ref string Msg),

в которой параметр Inpt — входное сообщение (запрос); параметр Conf — конфигурационные данные в форме массива строк вида "имя = значение"; строковый параметр Msg на входе принимает значение опционального параметра функции, а возвращает диагностическое сообщение функции. С этого момента все библиотечные функции, соответствующие данной спецификации, начинают играть роль сервисных функций, доступных для клиентских программ. Все остальные функции попросту игнорируются.

2. Переход от последовательной обработки к параллельной

В новой версии PMS переход от строго последовательной обработки к параллельно-последовательной был выполнен по следующим принципам.

- Сервисные функции получают и возвращают не один объект класса PmsMessage, а произвольный массив таких объектов. Другими словами, спецификация сервисной функции в новой версии выглядит так:

PmsMessage [] *имя_функции* (PmsMessage [] Inpt, string [] Conf, ref string Msg)

(разумеется, и входной и выходной массивы объектов PmsMessage могут по-прежнему включать только один элемент, но могут включать и несколько).

- В структуру конвейерной строки вводится разметка групп параллельно и последовательно выполняющихся элементов с помощью квадратных и фигурных скобок соответственно.

В качестве примера рассмотрим следующую конвейерную строку:

"Lib1.Func1, [Srv2/Lib2.Func2, {Lib3.Func3, Lib4.Func4, Lib5.Func5}, Srv3/(Lib6.Func6, Lib7.Func7)], Lib8.Func8"

Принципы обработки этой конвейерной строки в сервере PMS проиллюстрированы на рис. 2. Последовательно выполняемые элементы разнесены по горизонтали, а параллельно выполняемые — по вертикали.

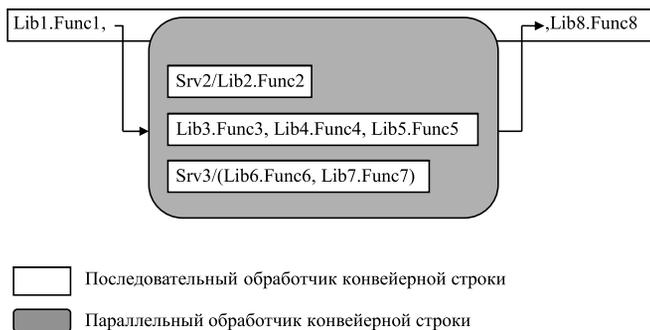


Рис. 2. Параллельно-последовательная обработка конвейерной строки

Как видно из рис. 2, три группы элементов конвейерной строки, которые были выделены квадратными скобками, выполняются параллельно. Первая группа включает единственный элемент, который обозначает вызов обрабатываемой функции на удаленном сервере Srv2. Вторая группа включает последовательный вызов трех функций непосредственно на обрабатываемом сервере (в данном случае группировка выполнена с помощью фигурных скобок). Наконец, третья группа подразумевает последовательное выполнение двух обрабатываемых функций на удаленном сервере Srv3 (конвейерная строка "Lib6.Func6, Lib7.Func7" целиком передается на выполнение в удаленный сервер). Отметим, что на вход первой сервисной функции в каждой группе будет передан один и тот же объект класса PmsMessage, сформированный сервисной функцией Lib1.Func1, а на вход функции Lib8.Func8 будет передан массив из трех таких объектов, сформированных сервисными функциями всех трех параллельно выполненных групп элементов.

Логика обработки конвейерной строки сосредоточена в двух программных классах: Последовательный обработчик конвейерной строки (ПСО) и Параллельный обработчик конвейерной строки (ПРО). При обработке каждого обращения от клиента сервер PMS сразу же создает экземпляр класса ПСО и передает ему на обработку полученную конвейерную строку вместе с полученным массивом защищенных сообщений (объектов класса PmsMessage). ПСО обеспечивает последовательное выделение каждого элемента из конвейерной строки и инициирует его обработку — вызов соответствующей сервисной функции из собственных библиотек или же обращение к удаленному серверу.

Если ПСО обнаруживает группу параллельно выполняемых элементов в конвейерной строке, то он полностью выделяет ее, созда-

ет экземпляр класса ПРО и передает ее этому объекту на выполнение вместе с массивом объектов PmsMessage, сформированным последней выполненной сервисной функцией. Логика работы ПРО включает:

- выделение всех параллельно выполняемых элементов из полученной конвейерной подстроки;
- создание отдельного экземпляра класса ПСО для каждого выделенного элемента и создание параллельных программных нитей (поток) для организации одновременной работы каждого из экземпляров;
- запуск всех созданных программных нитей с передачей каждой соответствующего элемента конвейерной строки и массива объектов класса PmsMessage, полученного от первичного ПСО;
- ожидание конца работы всех созданных программных нитей, объединение результатов их работы в единый массив объектов класса PmsMessage и возврат его в ожидающий ПСО для продолжения обработки (очень важно, что результирующий массив формируется не в порядке завершения нитей, но в порядке, заданном в конвейерной строке).

Подчеркнем, что "вторичные" экземпляры класса ПСО, созданные для обработки параллельно выполняемых элементов и последовательностей элементов, ничем не отличаются от "первичного", созданного при обнаружении обращения от клиента. Это, в частности, означает, что при обнаружении группы параллельно выполняемых элементов в обрабатываемой конвейерной подстроке "вторичный" экземпляр класса ПСО создаст, в свою очередь, новый экземпляр класса ПРО и организует взаимодействие с ним по вышеуказанным правилам.

3. Сетевой протокол PMS

Взаимодействие между клиентом и сервером PMS осуществляется по специальному, достаточно простому PMS-протоколу, ориентированному на передачу двоичных сетевых сообщений (PMS-сообщений) в обоих направлениях. Каждое такое сообщение в общем случае содержит два массива байтов: заголовок сообщения и тело сообщения (рис. 3). Первые четыре байта заголовка или тела сообщения содержат целое число — его длину. При передаче запроса от клиента к серверу в заголовок сетевого сообщения помещается строка, содержащая конвейерную строку, а в тело сообщения упа-

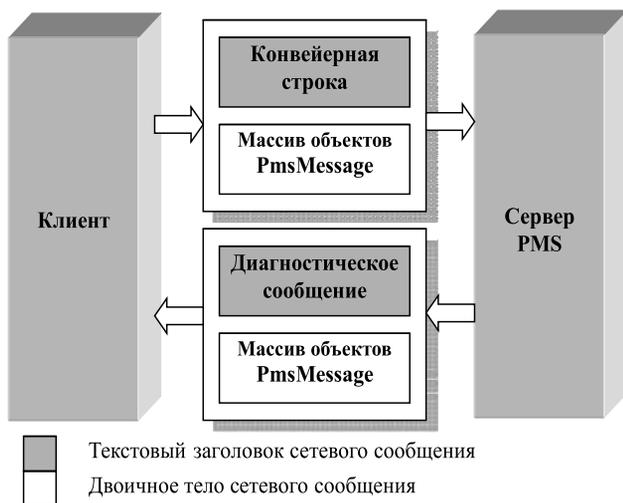


Рис. 3. Сетевые сообщения

ковывается массив объектов класса PmsMessage в открытой или зашифрованной форме, содержащий информационный запрос. Строка заголовка используется сервером для организации вызова сервисных обрабатывающих функций. При передаче результата обработки от сервера к клиенту в заголовок сетевого сообщения помещается строка диагностического сообщения, а в тело сообщения упаковывается массив объектов класса PmsMessage, содержащий ответ сервера в открытой или зашифрованной форме, предварительно подписанный собственным закрытым ключом сервера.

PMS в полной мере использует двоичную природу TCP/IP [10, 11]: никакие двоично-текстовые преобразования (типа base64) не применяются. Полученное от сервера диагностическое сообщение автоматически присваивается члену ErrMsg объекта класса PmsConnection на стороне клиента (см. рис. 1).

4. Временные оценки

Эксперименты с новой версией PMS проводились в лабораторной локальной сети (100 Мбит/с) с использованием нескольких четырехъядерных серверов с тактовой частотой 2,4 ГГц и операционной средой Window Server 2008, а также одноядерной клиентской рабочей станции с тактовой частотой 2,8 ГГц. Цель экспериментов заключалась в подтверждении корректности вышеописанных принципов организации сетевой службы, а также в получении количественных оценок того ускорения обработки, которое могут дать новые средства распараллеливания в условиях применения средств криптозащиты.

Рассмотрим один из таких экспериментов. Предположим, что у нас имеются четыре сервера PMS с именами Gun, Prsr, Prsf и Prsi, причем сервер Gun связан с базой данных, хранящей сведения об охотничьем оружии, а серверы Prsr, Prsf и Prsi связаны с базами данных, хранящими сведения о физических лицах, объявленных в региональный розыск, федеральный розыск и международный розыск соответственно. Допустим также, что

- в библиотеке Lb на сервере Gun имеется сервисная функция Owner, которая получает во входном сообщении серийный номер оружия и возвращает в выходном сообщении полное имя и дату рождения владельца одной строкой;
- в библиотеке Lb на серверах Prsr, Prsf и Prsi имеется сервисная функция FindPerson, которая получает в первой строке входного сообщения полное имя и дату рождения физического лица в качестве условий поиска, отыскивает в базе данных детальные сведения о физических лицах (гражданство, реквизиты документов и т. п.), отвечающих этим условиям, добавляет в конец сообщения эти сведения построчно и возвращает результат в качестве выходного сообщения.

Если в данных условиях клиентская программа поместит серийный номер оружия в защищенное сообщение (объект класса PmsMessage), сформирует для него одну или несколько электронных подписей, установит соединение с сервером Gun и передаст сообщение на обработку с помощью метода "Process", задав следующую конвейерную строку:

```
"Lb.Owner, Prsr/Lb.FindPerson, Prsf/Lb.FindPerson, Prsr/Lb.FindPerson",
```

то в результате последовательной обработки в четырех серверах он получит защищенное сообщение, содержащее результат: детальные сведения о всех разыскиваемых лицах, которые могут иметь отношение к оружию с заданным серийным номером.

Очевидно, что поиск физических лиц в трех базах данных можно выполнять параллельно, а не последовательно. Для использования средств распараллеливания новой версии PMS нам понадобится еще одна сервисная функция на сервере Gun — функция Join, которая получает на входе массив защищенных сообщений и формирует одно выходное сообщение, содержащее объединение (конкатенацию) данных, содержащихся во входном массиве. Если повто-

речь эксперимент с использованием конвейерной строки

"Lb.Owner, [Prsr/Lb.FindPerson, Prsf/Lb.FindPerson, Prsi/Lb.FindPerson], Lb.Join",

то клиент получит ту же информацию, что и в предыдущем случае (и также в единственном защищенном сообщении).

На рис. 4 отображены оценки времени обработки в миллисекундах для двух рассмотренных экспериментов в форме диаграммы (белый и черный столбики) в двух режимах: при применении только ЭЦП и при применении ЭЦП вместе с шифрованием данных на основе криптосистемы "КриптоПро CSP" версии 3.6. В обоих экспериментах время операции поиска владельца паспорта в базе данных оружия по его серийному номеру составляло 0,1 с, а время поиска детальной информации о физических лицах в базах данных физических лиц — 0,3 с.

Из рассмотренного примера видно, что распараллеливание может дать значительный выигрыш в скорости при обработке одиночных

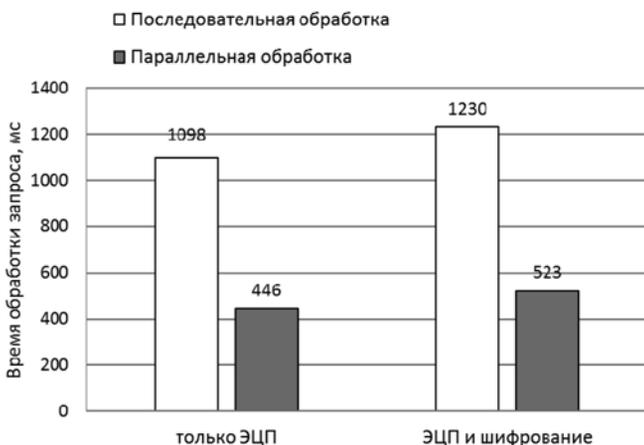


Рис. 4. Пример соотношения скоростей обработки

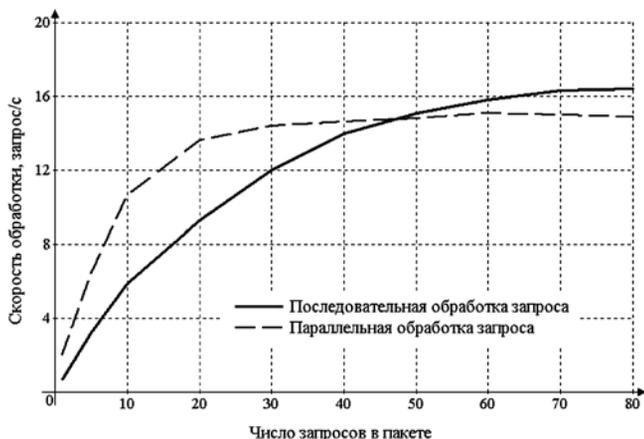


Рис. 5. Скорость обработки пакетов одновременных запросов

запросов. Однако на том же примере можно убедиться, что при высокой нагрузке на сервер соотношение скоростей обработки может быть более сложным.

Сервер PMS имеет многоканальную организацию, т. е. он создает по крайней мере один отдельный обрабатывающий программный поток ("нить") для обслуживания каждого запроса. На рис. 5 приведены результаты экспериментальной оценки быстродействия PMS в условиях высокой нагрузки: при параллельной обработке пакетов одновременно поступивших информационных запросов с применением криптозащиты (ЭЦП и шифрования). Кривые, представленные на рис. 5, отражают характерную зависимость скорости обработки запросов от их числа в пакете для рассмотренного выше примера. Первая из кривых соответствует строго последовательной обработке каждого отдельного запроса (т. е. без использования квадратных скобок в конвейерной строке), а вторая — параллельной обработке. Другими словами, вторая кривая отражает эффект от сочетания двух уровней распараллеливания: многопоточное выполнение запросов в пакете дополняется параллельным выполнением каждого отдельного запроса. Скорость обработки для обеих кривых вычислялась как частное от деления числа запросов в пакете на полное время его выполнения.

Как видно из рис. 5, обе кривые ведут себя в целом одинаково. При увеличении числа запросов в пакете растет и скорость обработки, что объясняется положительным эффектом от многопоточной обработки запросов в сервере PMS. Например, при десяти запросах в пакете скорость обработки достигает 5,7 запросов в секунду в случае последовательного выполнения отдельного запроса и 10,8 — в случае параллельного. При двадцати запросах в пакете значения производительности составляют 9,3 и 13,6 соответственно. При дальнейшем увеличении размеров пакета рост скорости обработки замедляется, а потом и вовсе останавливается вследствие достижения предельной производительности.

Однако на этом сходство в поведении кривых заканчивается. Как видно из рис. 5, при небольших размерах пакета параллельная обработка запроса дает значительный эффект в сравнении с последовательной, но по мере увеличения размера этот эффект уменьшается, а предельное значение у второй кривой оказывается даже меньшим, чем у первой. По-видимому, это объясняется тем, что при большом числе одновременных запросов накладные расходы на управление параллельными

нитями и издержки конкуренции между ними оказываются слишком высокими. В этих условиях средства распараллеливания обработки запроса (включая и добавочную сервисную функцию Lb.Join) не приводят к росту производительности, но становятся скорее лишним обременением для сервера.

Хотя приведенные оценки относятся к конкретному примеру, они позволяют сформулировать следующие выводы.

- Выигрыш в быстродействии от применения параллельной обработки в новой версии PMS может быть весьма значительным (т. е. может измеряться в разгах, а не в процентах).
- Как в режиме одиночных запросов, так и в режиме пакетов запросов средства параллельной обработки в новой версии PMS могут позволить существенно повысить скорость обработки в мультисерверной среде. Однако при повышении нагрузки на сервер PMS эффект от применения этих средств уменьшается и может вовсе исчезнуть (рис. 5).

Заключение

Идея использования совокупной вычислительной мощности множества сетевых узлов для ускорения обработки данных была реализована в ряде универсальных систем распараллеливания вычислений в сети, таких как Nadoor, Spark или Disco [12–15]. Эти системы базируются на технологиях распределенных сетевых файловых систем и организованы согласно принципу "лучше экспортировать вычисления, чем данные". В данной работе автор попытался показать, что схожая идея может быть реализована и в рамках одной сетевой службы для решения более узкой задачи: организации параллельной обработки информационных запросов на множестве серверов в рамках традиционной архитектуры данных (обычные файловые системы и/или СУБД) в распределенных информационных системах.

Новая версия сетевой службы была реализована в форме двух дополняющих друг друга программных продуктов: сервера PMS (в форме постоянно активной Windows-службы) и библиотеки функций PmsBase.dll для клиентских приложений и библиотечных сервисных функций. Оба продукта реализованы на языке C# в среде Microsoft Visual Studio для среды MS Framework 4.0. Первые опыты ее использования и лабораторные эксперименты показали ее способность существенно сократить

трудозатраты разработчиков на организацию распределенной обработки и защиты данных в мультисерверной среде.

Список литературы

1. **Шапошников И. В.** Web-сервисы Microsoft.NET. СПб: БХВ-Петербург, 2002. 336 с.
2. **Мак-Дональд М., Шпуншта М.** Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
3. **Козлов А. Д., Орлов В. Л.** Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. М.: ИПУ РАН, 2017. 156 с.
4. **Салимова Ш. А.** Кибербезопасность в России: актуальные угрозы и пути обеспечения в современных условиях // Достижения вузовской науки 2021: сб. статей XVII Международного науч.-иссл. конкурса, Пенза, 20 января 2021 года. Пенза: Наука и Просвещение, 2021. С. 207–214.
5. **Жаранова А. О., Птицына Л. К.** Анализ влияния распределенности на качество функционирования комплексных систем защиты информации // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020): Сб. науч. статей IX Междунар. науч.-техн. и науч.-метод. конф. СПб: СПбГУТ, 2020. С. 324–327.
6. **Згоба А. И., Маркелов Д. В., Смирнов П. И.** Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30–38.
7. **Асратян Р. Э.** Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. 2016. № 11. С. 490–497.
8. **Артемов И. Ю.** Использование параллельной обработки данных для оптимизации работы программного обеспечения // Программные продукты и системы. 2020. № 3. С. 471–475.
9. **Golubtsov P.** Scalability and Parallelization of Sequential Processing: Big Data Demands and Information Algebras // Advances in Intelligent Systems and Computing. 2020. Vol. 1127 AISC. P. 274–298.
10. **Снейдер Й.** Эффективное программирование TCP/IP. Библиотека программиста. СПб.: Символ-Плюс, 2002. 320 с.
11. **Хант К.** TCP/IP. Сетевое администрирование. СПб.: Питер, 2007. 816 с.
12. **Уайт Т.** Nadoor: Подробное руководство. СПб.: Питер, 2013. 672 с.
13. **Любанович Б.** Простой Python. Современный стиль программирования. СПб.: Питер, 2016. 480 с.
14. **Prajapati M, Patel S.** A Review on Big Data with Data Mining // Lecture Notes on Data Engineering and Communications Technologies. 2021. Vol. 52. P. 155–160.
15. **Klymash M., Hordiichuk-Bublivska O., Tchaikovskiy I., Deschynskiy Y.** Modeling and Research of Processing Big Data Sets in Distributed Information Systems // Proceedings — 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020. Lviv-Slavske: IEEE, 2020. P. 858–863.

A Service for Parallel Processing of Protected Information Queries in a Multi-Server Environment

The principles of organizing an Internet service designed to implement parallel processing of protected queries in distributed information systems that are oriented to work in complex network environments with many processing servers are considered. Distinctive feature of the service consists in a possibility to combine sequential ("pipelined") and parallel query processing in a multi-server environment. The service architecture is based on the concept of a "Protected message" corresponding to a container for electronic documents (information query or response) that can be provided with one or more electronic digital signatures. In contrast, for example, to the Web services technology, the described service is based not on the model of calling methods of remote objects, but on the message exchange model. In this case, this means that all service processing functions (methods) have the same strict specification: they receive an array of objects of the "Protected message" program class as a parameter and return an array of objects of the same class. In sequential processing, both arrays usually contain one "Protected message" object each. However, when using parallel processing, arrays can contain multiple elements that correspond to the results of processing by multiple software threads. These processing functions are grouped into one or more dynamic libraries, which are connected to the PMS server when it starts working (each library can be considered as a remote analogue of a Web service), and become available to clients.

Keywords: distributed systems, network technologies, Web technologies, parallel computing, network interaction, information security

DOI: 10.17587/it.27.592-599

References

1. **Shaposhnikov I. V.** Web-services Microsoft.NET, SPb., BHV-Peterburg, 2002, 336 p. (in Russian).
2. **Mak-Donald M., Szpuszta M.** Pro ASP.NET 3.5 in C#2008 Includes Silverlight 2, Moscow, Viljams, 2009, 1408 p. (in Russian).
3. **Kozlov A. D., Orlov V. L.** Methods and tools for ensuring information security of distributed corporate systems, Moscow, IPU RAN, 2017, 156 p. (in Russian).
4. **Salimova S. A.** Cybersecurity in Russia: current threats and ways to ensure in modern conditions, *Dostizhenija vuzovskoj nauki 2021: sbornik statej XVII Mezhdunarodnogo nauchno-issledovatel'skogo konkursa, Penza, 20 janvarja 2021 goda*, Penza, Nauka i Prosveshhenie, 2021, pp. 207–214 (in Russian).
5. **Zharanova A. O., Pticyna L. K.** Analysis of the impact of distribution on the quality of functioning of complex information security systems, *Aktual'nye problemy infotelekkommunikacij v nauke i obrazovanii (APINO 2020): Sbornik nauchnyh statej IX Mezhdunarodnoj nauchno-tehnicheskoi i nauchno-metodicheskoi konferencii*, SPb., SPBGUT, 2020, pp. 324–327.
6. **Zgoba A. I., Markelov D. V.** Cyber security: threats, challenges, decisions, *Voprosy Kiberbezopasnosti*, no. 5, 2014, pp. 30–38 (in Russian).
7. **Asratian R. E.** Internet service for protected information queries processing in distributed systems, *Programmaja Inzhenerija*, no. 11, 2016, pp. 490–497 (in Russian).
8. **Artemov I. Y.** Using parallel data processing to optimize software performance, *Programmnye Produkty i sistemy*, 2020, no 3, pp. 471–475 (in Russian).
9. **Golubtsov P.** Scalability and Parallelization of Sequential Processing: Big Data Demands and Information Algebras, *Advances in Intelligent Systems and Computing*, 2020, vol. 1127 AISC, pp. 274–298.
10. **Snader J.** Effective TCP/IP programming, SPb., Simvol-Pljus, 2002, 320 p. (in Russian).
11. **Hant K.** TCP/IP. Network administration, SPb., Piter, 2007, 816 p. (in Russian).
12. **Уайт Т.** Hadoop: The definitive guide, SPb., Piter, 2013, 672 p. (in Russian).
13. **Lubanovich B.** Simple Python. Modern programming style. SPb., Piter, 2016. 480 p. (in Russian).
14. **Prajapati M, Patel S.** A Review on Big Data with Data Mining, *Lecture Notes on Data Engineering and Communications Technologies*, 2021, vol. 52, pp. 155–160.
15. **Klymash M., Hordiichuk-Bublivska O., Tchaikovskiy I., Deschynskiy Y.** Modeling and Research of Processing Big Data Sets, *Distributed Information Systems, Proceedings — 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020, Lviv-Slavske, IEEE*, 2020, pp. 858–863.

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 616-073.756.8-027.44: 004.93

DOI: 10.17587/it.27.600-606

В. С. Осипович¹, канд. техн. наук, доц., e-mail: seth22@yandex.ru,
О. С. Медведев¹, магистр техн. наук, инженер, e-mail: o.med@bsuir.by;,
О. Н. Дудич², канд. мед. наук, доц., e-mail: oksana_s20@mail.ru,
В. Л. Краси́льникова², д-р мед. наук, проф, e-mail: seth22@mail.ru,
К. Д. Яшин¹, канд. техн. наук, доц., e-mail: yashin@bsuir.by,

¹Белорусский государственный университет информатики и радиоэлектроники, г. Минск,

²Белорусская медицинская академия последипломного образования, г. Минск

Технология преобразования результатов компьютерной томографии в трехмерные модели

Разработка относится к информационным технологиям и может быть использована для обработки медицинских изображений. Поставленная задача преобразования результатов компьютерной томографии в трехмерные модели достигается тем, что DICOM-файлы, содержащие результаты компьютерной томографии головы человека, подвергаются корректировке. В них меняют цвет пикселей в местах перехода от черного к серому или наоборот, а также заменяют черный пиксель на белый. В результате достигается сокращение процента потери информации о мелких костях с 22...31 % до 3...5 %, а следовательно, достигается увеличение точности трехмерной модели лицевых костей черепа человека. Все это способствует существенному улучшению качества лицевой хирургии.

Ключевые слова: преобразование результатов компьютерной томографии, DICOM-изображения, 3D-модель

Введение

Посттравматические дефекты и деформации нарушают привычную геометрию лица, приводят к развитию функциональных нарушений. В медицинскую практику внедряются современные компьютерные технологии производства персонализированных титановых имплантатов для устранения дефектов костей черепа [1–3]. Использование персонализированных имплантатов в реконструктивно-восстановительной хирургии лица продемонстрировало их неоспоримые достоинства, которые заключаются в высокой точности, уменьшении травматичности, сокращении длительности операции и, в конечном счете, в достижении предсказуемого стабильного функционального и косметического результата [4].

Для создания высокоточных персонализированных имплантатов определяющее значение имеет качество микроспиральной компьютерной томографии (МСКТ). Анализ полученных в результате МСКТ изображений [5–7] в формате DICOM [8] с последующей трехмер-

ной реконструкцией черепа дает возможность хирургу более достоверно оценить анатомические особенности лица пациента [9, 10].

Однако среди костей лицевого скелета нижняя и медиальная стенки глазницы являются очень тонкими структурами (толщина примерно 0,1...0,3 мм). Это меньше разрешающей способности существующих аппаратов (до 0,625 мм). В связи с этим построение 3D-моделей с использованием программного обеспечения (ПО) томографа не дает полной информации о состоянии нижней и медиальной стенок глазницы. На снимках они выглядят как ячеистая структура, имеющая поры. В результате преобразования файлов DICOM в 3D-модель теряется часть информации о тонких и мелких костях глазницы. Эти потери приводят к сложностям в определении геометрии поврежденных костей. Также следует отметить, что программное обеспечение существующих МСКТ не позволяет редактировать полученные 2D-изображения.

Известно программное средство 3D Sliser. Оно обеспечивает открытие и просмотр DICOM-файлов, а также конвертацию их в трехмерную

модель с сохранением в формате STL. Технология заключается в том, что набор изображений в формате DICOM преобразуют в массив с элементами, несущими информацию о пространственном расположении пикселя и его цвете. Затем формируют ячейки размером $2 \times 2 \times 2$ элементов массива. После этого осуществляют анализ элементов ячейки и формируют поверхность, отделяющую белые элементы массива от элементов остального цвета, в виде треугольника, ориентация которого в пространстве задается на основе таблицы шаблонов для ячейки. Координаты вершин треугольников используют для записи элементов поверхности в файл формата STL [11–14]. Ограничением этой технологии являются потери информации о тонких и мелких костях при конвертации.

Цель настоящего исследования — разработать алгоритм и технологию преобразования результатов компьютерной томографии в трехмерные модели лицевого черепа для снижения потерь информации о мелких костях. Для достижения цели решены следующие задачи: разработана технология преобразования результатов компьютерной томографии; разработан алгоритм обработки исходных DICOM-изображений.

Методика эксперимента

Изображения формата DICOM получены в результате МСКТ (Siemens Emotion 6, Германия) голов пациентов с повреждениями костей глазницы. Проанализированы данные пациентов обоих полов в возрасте от 20 до 40 лет, имеющих различные повреждения костей глазницы: от перелома одной кости, до более сложных переломов, затрагивающих верхнечелюстную кость.

DICOM-изображения отражают плотность исследуемых участков тела в виде среза: чем плотнее участок кости, тем насыщеннее белый цвет на изображении. Мягкие ткани отображены серым цветом, а пустоты имеют черный цвет. Анализ изображений трех проекций с использованием программного обеспечения 3D Slicer показал (рис. 1, см. четвертую сторону обложки), что тонкие кости глазницы дали поглощение рентгеновского луча меньше, чем должны давать кости. Они имеют на изображении DICOM, а также на остальных проекциях, цвет, отличный от белого.

Программное обеспечение 3D Slicer при построении трехмерной модели лицевых костей черепа фиксирует в качестве исходных данных

пиксели белого цвета. Получается, что при генерации трехмерной модели происходит потеря важной информации о костях глазницы. Потерю информации оценивали путем определения площади костей глазницы и суммы площадей дефектов (отверстий) в этих костях на трехмерной модели.

Таким образом, следует внести дополнения в исходные изображения в формате DICOM: необходимо сделать тонкие кости видимыми для алгоритмов программного обеспечения 3D Slicer. При этом нельзя внести искажения в результат построения 3D-модели костей головного черепа пациента. Должна сохраниться геометрия повреждения костей глазницы.

Анализ трех проекций показал, что тонкие кости глазницы граничат с пустотами в черепной коробке. В то же время при повреждении костей глазницы эти пустоты заполняются биологическими клетками, в основном жировыми клетками. Это приводит к изменению уровня поглощения рентгеновского излучения и изменению изображения DICOM относительно здоровой глазницы. Поэтому при разработке программного обеспечения, вносящего изменения в исходные DICOM-файлы, нужно и можно это учитывать.

Результаты и их обсуждение

Технология преобразования результатов компьютерной томографии. Разработанное программное средство функционирует следующим образом (рис. 2). Первый этап: программное средство загружает исходные DICOM-файлы и инициализирует пути их сохранения. Второй этап: программное средство осуществляет выбор и инициализацию значений пикселей белого цвета. Далее программное средство обрабатывает файлы по циклу, включающему этапы 3–7. Третий этап: программное средство открывает один из файлов комплекта. Четвертый этап: программное средство считывает пиксельные данные в массив. Пятый этап: программное средство осуществляет постолбцовый проход по массиву с присвоением черному пикселю значения белого на границе перехода

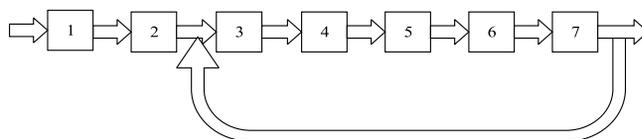


Рис. 2. Схема алгоритма работы программного средства

от черного к серому пикселям. Шестой этап: программное средство сохраняет массив пиксельных данных в DICOM-файл. Седьмой этап: программное средство проводит проверку на завершение обработки файлов (последний ли файл комплекта обработан либо следует повторить цикл для следующего файла). Информация о цвете пикселей в DICOM-файлах хранится в соответствии с форматом gray 16 [15–17]. Для определения черного, серого и белого цветов при разработке специального программного обеспечения был проанализирован спектр формата gray 16. В качестве критерия выбора при анализе спектра выступало определенное визуальное среднее число пикселей черного, белого и серого цветов, присущее областям на DICOM-изображении, отвечающим за пустоты, костную ткань, прочие ткани и жидкости тела человека, соответственно. Проверка выбора осуществлялась путем анализа DICOM-изображений после их обработки разработанным программным обеспечением.

Размерность массива пиксельных данных зависит от разрешения исходных файлов DICOM — ширины и длины изображения в пикселях, соответственно. Так как в gray 16 каждый пиксель состоит из двух байтов, значение старшего байта обозначили i , значение младшего байта — j . Тогда в качестве черного приняты следующие диапазоны значений i и j :

$$\begin{aligned} i &= 16n; \\ j &= [0; 255], \end{aligned}$$

где n — целое число от 0 до 15; i — номер элемента массива пикселей по горизонтали; j — номер элемента массива пикселей по вертикали.

В качестве белого цвета приняты следующие значения байтов пикселей формата gray 16:

$$\begin{aligned} i &= 12 + 16n; \\ j &= [144; 255]; \\ i &= [13; 15] + 16n; \\ j &= [0; 255]. \end{aligned}$$

Оставшиеся значения байтов пикселей формата gray16 приняты в качестве серого цвета.

Расчет оценки процента потерь информации при преобразовании из DICOM в STL осуществили путем расчета процента площади костей глазницы, исчезнувших при конвертации.

Для расчета ориентировочной площади глазницы в 3D-редакторе были созданы четыре треугольника, которые расположены в разных плоскостях глазницы (рис. 3, см. четвертую сторону обложки).

После измерения значений сторон всех треугольников были рассчитаны их площади с использованием формулы Герона[18–21]:

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \quad (1)$$

где S — площадь глазницы, $p = \frac{a+b+c}{2}$ — полупериметр и a, b, c — стороны треугольника.

Приблизительная общая площадь нижней и медиальной костей глазницы составила 995,97 мм².

Для нахождения площадей отверстий были использованы круги и прямоугольники, которые вписывались в отверстия трехмерной модели (рис. 4, см. четвертую страницу обложки).

Для расчета площади треугольников была использована формула (1), а для расчета площадей прямоугольников и кругов — формулы

$$S = ab; \quad (2)$$

$$S = \pi r^2, \quad (3)$$

где a — ширина прямоугольника; b — длина прямоугольника; r — радиус круга.

Сложив данные значения, получаем площадь отверстий, равную 233,72 мм². Таким образом, процент потерь информации о костях составляет 23,467 %.

Аналогичные расчеты процента потери информации для трехмерной модели, полученной разработанным программным средством, дают значение 3,8 %.

В общей сложности расчет процента потери информации для конвертации DICOM-изображений в STL-формат с применением разработанной технологии и без него был осуществлен для 30 комплектов. Было установлено, что потери информации с применением данного способа составили 3...5 %, а без него — 22...31 %.

Блок-схема алгоритма обработки исходных DICOM-изображений. Результат построения трех проекций инструментами 3D Slicer на основании DICOM-изображений, обработанных специальным программным обеспечением, отражен на рис. 5.

В сравнении с рис. 2 видно следующее. Места, отмеченные на рис. 5 как фрагменты с неявной костью, преобразованы в явный белый цвет, характеризующий наличие костной ткани. Однако обнаружен существенный недостаток обработки DICOM-изображений разработанным программным обеспечением. К сожалению, алгоритм преобразует в костную ткань кожу

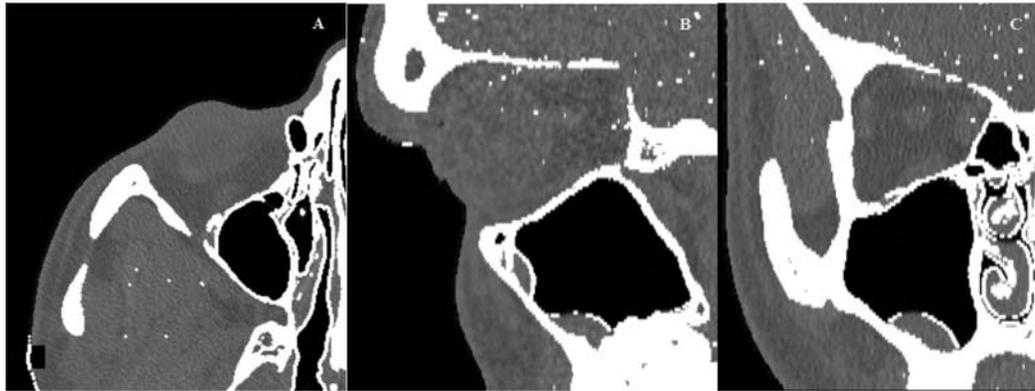


Рис. 5. Три проекции головы пациента, построенные на основе преобразованных специальным программным обеспечением исходных DICOM-изображений

на лице пациента. Это усложняет просмотр и дальнейшую работу по анализу и построению геометрии повреждения (для дальнейшего изготовления индивидуального импланта). В целях устранения этого недостатка были внесены изменения в алгоритм обработки специального программного обеспечения: были добавлены дополнительные действия по корректировке массива пиксельных данных (рис. 6).

Исходные результаты реконструкции лицевых костей черепа в виде трехмерной модели изображены на рис. 7. В результате обработки исходных DICOM-изображений модифицированным специальным программным обеспечением и выполнения операций 2—4 предложенной технологической цепочки получена трехмерная модель костей лицевого черепа (рис. 7). При анализе изображения на рис. 7 видно, что левая глазни-

ца трехмерной модели (рис. 7, а) имеет четкие, без отверстий, кости глазницы, в отличие от исходной трехмерной модели (рис. 7, б). Это позволяет однозначно идентифицировать глазницу с поврежденными костями. В дальнейшем, используя редакторы 3D-моделирования, возможно осуществлять построение трехмерной модели костей глазницы.

Расчет потерь информации о костях глазниц при построении трехмерной модели лицевых костей черепа показал, что при использовании DICOM-изображений без предварительной обработки среднее значение потерь составляет 22...31 % при шаге сканирования 0,625 мм, а при использовании DICOM-изображений после обработки разработанным программным обеспечением — 3...5 %.

Кроме того, проанализированы четыре варианта построения проекции и трехмерных моделей. Варианты отличались числом пикселей, преобразуемых специальным программным обеспечением в белый цвет (от одного до четырех пикселей). Результат оценки процента потери информации о костях глазницы в зависимости от числа преобразуемых пикселей отражен в таблице.

Таким образом, анализ вариантов построения трех проекций и трехмерных моделей на основании обработанных специальным программным обеспечением изображений показал,

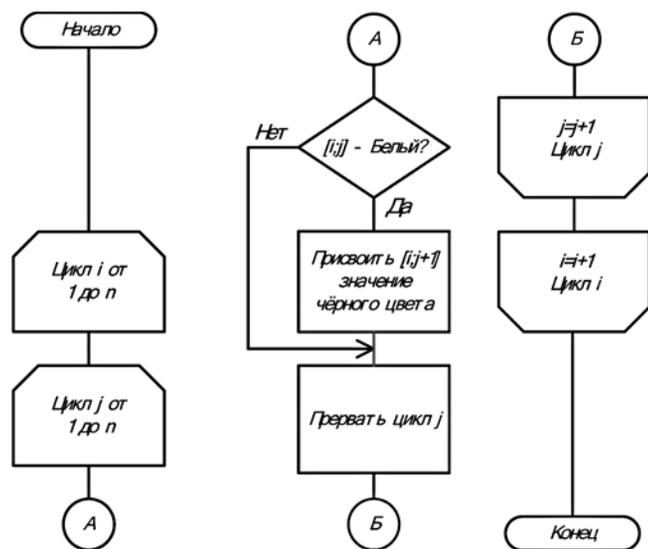


Рис. 6. Блок-схема алгоритма обработки DICOM-изображений

Результат оценки потери информации о костях глазницы в зависимости от числа преобразуемых пикселей

Оценка	Число преобразуемых пикселей			
	1	2	3	4
Потери информации о костях глазницы, %	4...7	3...5	3...5	3...5

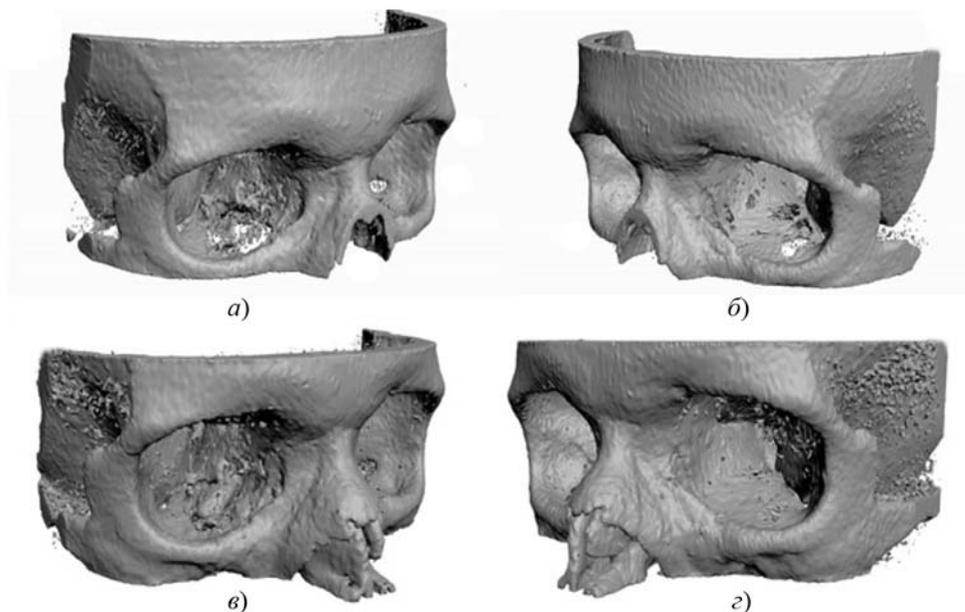


Рис. 7. Вид трехмерной модели лицевых костей черепа после обработки специальным программным обеспечением — результат построения 3D-редактором:

a — вид реконструкции правой глазницы; *б* — вид реконструкции левой глазницы; *в, г* — результаты реконструкции лицевых костей черепа после обработки программным средством

что для получения трехмерной модели с сохранением полной информации о структуре костей глазницы достаточно преобразования в белый цвет (цвет костной ткани) двух пикселей.

Сопоставительный анализ с известными техническими решениями указывает на то, что разработанная авторами технология отличается предварительной обработкой изображения в формате DICOM. Технология позволяет обеспечить сохранение информации о тонких и мелких костях глазницы при конвертации срезов в трехмерную модель.

Заключение

Установлено, что для построения трехмерной модели с сохранением полной информации о структуре костей глазницы требуется преобразование минимум двух пикселей в цвет костной ткани при предварительной обработке специальным программным обеспечением.

Установлено, что обработка исходных DICOM-изображений разработанным специальным программным обеспечением приводит к снижению потери информации о костях глазницы при построении трехмерной модели лицевых костей черепа с 22...31 % до 3...5 %, т. е. может быть осуществлена коррекция DICOM-изображений в автоматическом режиме.

Представлена технология преобразования результатов компьютерной томографии в трехмерную модель, включающая:

- преобразование DICOM-файлов в массив с элементами, несущими информацию о пространственном расположении пикселя и его цвете;
- формирование ячейки размером $2 \times 2 \times 2$ элементов массива;
- анализ элементов ячейки и формирование поверхности, отделяющей белые элементы массива от элементов остального цвета, в виде треугольника, ориентация которого в пространстве задается на основе таблицы шаблонов для ячейки;
- запись элементов поверхности в файл формата STL, отличающийся тем, что исходные DICOM-файлы подвергаются предварительной обработке путем замены черного пикселя на белый в местах перехода от черного пикселя к серому.

Сущность предложенной технологии заключается в предварительной обработке DICOM-файлов по следующему алгоритму: изменяется цвет пикселей в местах перехода от черного к серому — заменяется черный пиксель на белый. Далее набор изображений в формате DICOM преобразуется в массив с элементами, несущими информацию о пространственном расположении пикселя и его цвете. Затем формируются ячейки размером $2 \times 2 \times 2$ элементов

массива. После этого осуществляется анализ элементов ячейки и формируется поверхность, отделяющая белые элементы массива от элементов остального цвета, в виде треугольника, ориентация которого в пространстве задается на основе таблицы шаблонов для ячейки. Координаты вершин треугольников используются для записи элементов поверхности в файл формата STL.

Список литературы

1. Шаробаро В. И., Потапов А. А., Гаврилов А. Г., Мантурова Н. Е., Кравчук А. Д., Еолчийн С. А., Иванов Ю. В., Латышев Я. А., Чобулов С. А. Выбор метода реконструкции при обширных комбинированных дефектах мозгового черепа в зависимости от клинической ситуации // *Анналы пластической, реконструктивной и эстетической хирургии*. 2018. № 4. С. 3—14.
2. Стучилов В. А., Никитин А. А., Секирин А. Б., Филатова Е. В., Ларионов К. С. Компьютерное моделирование и лазерные технологии в реабилитации пациентов при травме средней зоны лица // *Альманах клинической медицины*. 2015. Т. 36. С. 82—89.
3. Schramm A., Suarez-Cunqueiro M. M., Rücker M., Kokemüller H., Bormann K. H., Metzger M. C., Gellrich N. C. Computer-assisted therapy in orbital and mid-facial reconstructions // *Int. J. Med. Robotics Comput. Assist. Surg.* 2009. N. 5. P. 111—124.
4. Davydov D. V., Levchenko O. V., Mikhaylyukov V. M. Surgical reconstruction of posttraumatic defects and deformities of the orbit using frameless navigation // *Вестник офтальмологии*. 2014. № 2. С. 20—26.
5. Masalitina N. N., Kurochka K. S. The automated classifier of computer tomography results for lumbar spine geometric modeling // *Doklady BGUIR*. 2017, Vol. 105, N. 3. P. 12—19.
6. Patent 2000125015/14 RF A61B 17/80, A61B 17/56, A61F 2/28. Rybalchenko's plate for reconstructing damaged bone structures of cheekbone, eye-socket and upper jaw complex and the way of reconstructing damaged bone structures of cheekbone, eye-socket and upper jaw complex with this plate. Rybalchenko G. N.; Closed Joint Stock Company "KONMET Incorporated", Rybalchenko G. N. Application № 2202302, 04.10.2000; published 20.04.2003.
7. Patent 2000116451/14 RF A61 F2/28. Method manufacturing an individual precision implant for compensation of complex subtotal polyostotic eye-socket defect / Shalumov A. Application № 2164392, 27.06.2000; published 27.03.2001.
8. Стандарт DICOM 3.0. 2016. URL: <http://www.courseas.ru/dicomdoc.html> (дата доступа: 18.02.2016).
9. Красильникова В. Л., Дудич О. Н., Залуцкий И. В. Обоснование выбора конструкции имплантата из титана, предназначенного для устранения костных дефектов стенок глазницы на основе морфогистологического исследования тканей // *Офтальмология. Восточная Европа*. 2017. № 1. С. 80—102.
10. Pgm — Netpbm grayscale image format. 2016. URL: <http://netpbm.sourceforge.net/doc/pgm.html> Дата доступа: 6.02.2018.
11. Стучилов В. А., Никитин А. А., Секирин А. Б., Филатова Е. В., Ларионов К. С., Гришин А. С., Кокорев В. Ю., Рябцева А. А. Компьютерное моделирование и лазерные технологии в реабилитации пациентов при травме средней

зоны лица // *Альманах клинической медицины*. 2015. Т. 36. С. 82—89.

12. Hussain T. ViPS: A novel visual processing system architecture for medical imaging // *Biomedical Signal Processing and Control*. 2017. Vol. 38. P. 293—301.

13. Mohana G., Subashini M. MRI based medical image analysis: Survey on brain tumor grade classification // *Biomedical Signal Processing and Control*. 2018. Vol. 39. P. 139—161.

14. Mitchell D. W. A Heron-type formula for the reciprocal area of a triangle // *Mathematical Gazette*. November 2005. N. 89. P. 494.

15. Asipovich V. S., Dudich O. N., Krasilnikova V. L., Karalko A. A., Radnionok A. L., Moroz P. A., Nikolaev A. Y., Konovalova M. A., Yashin K. D. Deep Learning in Processing Medical Images and Calculating the Orbit Volume // *ICNBME 2019: 4th International Conference on Nanotechnologies and Biomedical Engineering*, Chisinau, Moldova, 18 September 2019 — Part of the IFMBE Proceedings book series. Vol. 77. P. 519—522.

16. Дудич О. Н., Красильникова В. Л., Осипович В. С., Николаев А. Ю., Радненко А. Л., Яшин К. Д. Углубленный анализ больших данных инновационной лицевой хирургии // *Сборник материалов V международной научно-практической конференции "BIGDATA и анализ высокого уровня"* Республика Беларусь, Минск, 13—14 марта 2019 года. Минск: БГУИР, 2019. С. 54—68.

17. Radnionok A., Asipovich V., Dudzich A., Krasilnikova V., Nikolayev A., Yashin K. Algorithm of processing microspiral ct-scan results for constructing a three-dimensional model of orbit thin bones // *Journal of Engineering Science*. 2018. Vol. XXV, N. 4. P. 39—46.

18. Дудич О. Н., Красильникова В. Л., Осипович В. С., Николаев А. Ю., Радненко А. Л., Яшин К. Д. Компьютерное моделирование индивидуальных лицевых имплантов // *Доклады БГУИР*. 2018. № 7 (117). С. 106—111.

19. Осипович В. С., Николаев А. Ю., Радненко А. Л., Яшин К. Д. Система построения геометрических моделей лицевых имплантов // *Сборник материалов IV международной научно-практической конференции "BIG DATA and ADVANCED ANALYTICS"*, 3—4 мая 2018 г. Минск, БГУИР. С. 355—359.

20. Николаев А. Ю., Радненко А. Л., Осипович В. С., Яшин К. Д. Обработка больших массивов выходных файлов компьютерного рентгеновского томографа для реконструктивной лицевой хирургии // *Сборник материалов III международной научно-практической конференции "Third International Conference and Expo BIG DATA and ADVANCED ANALYTICS"*, 3—4 мая 2017 г. Минск, БГУИР. С. 238—241.

21. Osipovich V., Dudzich O., Krasilnikova V., Dzik S., Nikolaev A., Kovalenko U., Yashin K. Processing of large amounts of information for reconstructive facial surgery // *BIG DATA and Advanced Analytics. Optimising Business and IT: сборник материалов конференции*. Минск, Республика Беларусь, 15—17 июня 2016 г. Минск, БГУИР. 2016. С. 89—97.

22. Осипович В. С., Дудич О. Н., Яшин К. Д., Николаев А. Ю., Дик С. К., Медведев О. С. Технология получения 3D модели костей лицевого черепа // *Сборник научных статей IX международной научно-технической конференции "Медэлектроника — 2015. Средства медицинской электроники и новые медицинские технологии"*, 4—5 декабря 2015 г. Минск, БГУИР. С. 43—44.

V. S. Osipovich¹, Ph.D., Associate Professor, e-mail: v.osipovich@bsuir.by,
O. S. Medvedev¹, Master of Engineering, Graduate Student, e-mail: o.med@bsuir.by,
O. N. Dudzich², M. D., Associate Professor, Department of Ophthalmology,
V. L. Krasilnikova², M. D., Professor, Department of Ophthalmology,

K. D. Yashin¹, Head of the Department of Human Engineering and Ergonomics,

¹Belarusian State University of Informatics and Radioelectronics, Minsk, 220013, Republic of Belarus,

²Belarus Medical Academy of Postgraduate Education, Minsk, 220013, Republic of Belarus

Technology for Conversion of Computer Tomography Results into Three-Dimensional Models

The development relates to information technology and can be used to process medical images. This task is achieved by the fact that DICOM files containing the results of computed tomography of the human head are subject to correction. They change the color of the pixels at the transition points from black to gray or vice versa, and also replace the black pixel with white. As a result, the percentage of loss of information about small bones is reduced from 22-31 % to 3-5 %, and, therefore, the accuracy of the three-dimensional model of the facial skull of the human head is increased. All this contributes to a significant improvement in the quality of facial surgery.

Keywords: Transformation of computed tomography results, DICOM images, 3D model

DOI: 10.17587/it.27.600-606

References

1. Sharobaro V., Potapov A., Gavrilov A., Manturova E., Kravchuk A., Eolchiyan S., Ivanov Yu., Latyshev Ya., Chobulov S. Choosing a reconstruction method for extensive combined defects of the cerebral skull, depending on the clinical situation, *Annals of Plastic, Reconstructive and Aesthetic Surgery*, 2018, no. 4, pp. 3–14 (in Russian).
2. Stuchilov V., Nikitin A., Sekirin A., Filatova E., Larionov K. Computer modeling and laser technologies in the rehabilitation of patients with midface trauma, *Almanac of Clinical Medicine*, 2015, vol. 36, pp. 82–89 (in Russian).
3. Schramm A., Suarez-Cunquero M. M., Rucker M., Kokenmueller H., Bormann K. H., Metzger M. C., Gellrich N. C. Computer-assisted therapy in orbital and mid-facial reconstructions, *Int. J. Med. Robotics Comput. Assist Surg.*, 2009, no. 5, pp. 111–124.
4. Davydov D., Levchenko O., Mikhaylyukov V. Surgical reconstruction of posttraumatic defects and deformities of the orbit using frameless navigation, *Ophthalmology bulletin*, 2014, no 2, pp. 20–26.
5. Masalitina N., Kurochka K. The automated classificatory of computer tomography results for lumbar spine geometric modeling, *Reports BGUIR*, 2017, vol. 105, no. 3, pp. 12–19.
6. Patent 2000125015/14 RF A61B 17/80, A61B 17/56, A61F 2/28. Rybalchenko's plate for reconstructing damaged bone structures of cheekbone, eye-socket and upper jaw complex and the way of reconstructing damaged bone structures of cheekbone, eye-socket and upper jaw complex with this plate. Rybalchenko G; Closed Joint Stock Company "KONMET Incorporated", Rybalchenko G. Application No 2202302, 04.10.2000; published 20.04.2003.
7. Patent 2000116451/14 RF A61 F2/28. Method manufacturing an individual precision implant for compensation of complex subtotal polyostotic eye-socket defect. Shalunov A. Application No 2164392, 27.06.2000; published 27.03.2001.
8. Standard DICOM 3.0, available at: <http://www.courseas.ru/dicomdoc.html> (date of access: 18.02.2016) (in Russian).
9. Krasilnikova V., Dudich O., Zalutskiy I. Substantiation of the choice of a titanium implant designed to eliminate bone defects in the walls of the orbit on the basis of morphohistological examination of tissues, *Ophthalmology. Eastern Europe*, 2017, no. 1, pp. 80–102 (in Russian).
10. Pgm — Netpbm grayscale image format, available at: <http://netpbm.sourceforge.net/doc/pgm.html> (date of access: 6.02.2018).
11. Stuchilov V., Nikitin A., Sekirin A., Filatova E., Larionov K., Grishin A., Kokorev V., Ryabtseva A. Computer modeling and laser technologies in the rehabilitation of patients with midface trauma, *Almanac of Clinical Medicine*, 2015, vol. 36, pp. 82–89 (in Russian).
12. Hussain T. ViPS: A novel visual processing system architecture for medical imaging, *Biomedical Signal Processing and Control*, 2017, vol. 38, pp. 293–301.
13. Mohana G., Subashini M. MRI based medical image analysis: Survey on brain tumor grade classification, *Biomedical Signal Processing and Control*, 2018, vol. 39, pp. 139–161.
14. Mitchell D. W. A Heron-type formula for the reciprocal area of a triangle, *Mathematical Gazette*, November 2005, no. 89, p. 494.
15. Asipovich V., Dudich O., Krasilnikova V., Karakulko A., Radnionok A., Moroz P., Nikolaev A., Konovalova M., Yashin K. *ICNBME 2019: 4th International Conference on Nanotechnologies and Biomedical Engineering, Chisinau, Moldova, 18 September 2019 — Part of the IFMBE Proceedings book series*, IFMBE, 2019, vol. 77, pp. 519–522.
16. Dudich O., Krasilnikova V., Osipovich V., Nikolaev A., Radnenok A., Yashin K. *Collection of materials of the V international scientific and practical conference "BIGDATA and high-level analysis"*, Republic of Belarus, Minsk, March 13-14, 2019, Minsk, BSUIR, 2019, pp. 54–68 (in Russian).
17. Radnionok A., Asipovich V., Dudzich A., Krasilnikova V., Nikolayev A., Yashin K. *Journal of Engineering Science*, 2018, vol. XXV, no. 4, pp. 39–46.
18. Dudich O., Krasilnikova V., Osipovich V., Nikolaev A., Radnenok A., Yashin K. *Reports of BSUIR*, 2018, no. 7 (117), pp. 106–111 (in Russian).
19. Osipovich V., Nikolaev A., Radnenok A., Yashin K. *Collection of materials of the IV international scientific-practical conference "BIG DATA and ADVANCED ANALYTICS"*, May 3-4, 2018, Minsk, BSUIR, pp. 355–359 (in Russian).
20. Nikolaev A., Radnenok A., Osipovich V., Yashin K. *Collection of materials of the III international scientific-practical conference "Third International Conference and Expo BIG DATA and ADVANCED ANALYTICS"*, May 3–4, 2017, Minsk, BSUIR, pp. 238–241 (in Russian).
21. Osipovich V., Dudzich O., Krasilnikova V., Dzik S., Nikolaev A., Kovalenko U., Yashin K. *BIG DATA and Advanced Analytics. Optimising Business and IT: collection of conference materials*, Minsk, Republic of Belarus, June 15–17, 2016, Minsk, BSUIR, 2016, pp. 89–97.
22. Osipovich V., Dudich O., Yashin K., Nikolaev A., Dick S., Medvedev O. *Collection of scientific articles of the IX international scientific and technical conference "Medelectronics — 2015. Means of medical electronics and new medical technologies"*, 4 — December 5, 2015, Minsk, BSUIR, pp. 43–44 (in Russian).

В. М. Гриняк, д-р техн. наук, проф.,
Владивостокский государственный университет экономики и сервиса,
А. В. Шуленина, ст. преподаватель,
Дальневосточный федеральный университет

**Кластеризация данных траекторий морских судов
для планирования маршрутов
через акватории с интенсивным движением***

Статья посвящена проблеме обеспечения безопасности движения судов на морских акваториях. Одним из элементов организации движения в районах интенсивного судоходства является система установления путей движения судов. Она представляет собой набор ограничений, накладываемых определенной схемой движения судов, правилами движения, принятыми на конкретной акватории. В работе рассматривается задача планирования маршрута перехода судна через акватории с интенсивным движением. При движении через акватории с установленными путями планирование маршрута перехода судна следует осуществлять с учетом заданных ограничений. Возможным путем идентификации этих ограничений является выделение паттернов движения конкретной морской акватории из ретроспективной информации о ее трафике. Модельные представления такой задачи могут быть сформулированы на основе идеи о кластеризации параметров движения судов. В основу модели при решении задачи планирования маршрута положен поиск кратчайшего пути на взвешенном графе. Предлагается несколько способов построения такого графа: регулярная сетка вершин и ребер, слоистая сетка вершин и ребер, случайная сетка вершин и ребер, вершины и ребра на основе ретроспективных данных. Вес ребер предлагается задавать как функцию "желательности" того или иного курса судна для каждой точки акватории с учетом выявленных паттернов движения. В работе обсуждаются возможные методы кластеризации, делается выбор в пользу субтрактивной кластеризации, не требующей предварительного задания числа кластеров. Источником данных о трафике акватории могут служить сервисы Автоматической идентификационной системы. В работе приведены примеры планирования маршрутов перехода через Токийский залив и Сангарский пролив.

Ключевые слова: управление движением судов, система установления путей движения, планирование маршрута перехода, интенсивное движение, кластеризация, большие данные, алгоритмы на графах, автоматическая идентификационная система

Введение

Развитие методов и инструментов сбора, хранения и обработки больших массивов данных, сервисов Автоматической идентификационной системы (АИС), технологий е-навигации дало возможность использования новых мето-

дов для решения классических задач судовождения, постановки и исследования новых задач. Одной из таких новых задач является оценка характерных паттернов движения морской акватории, методы решения которой основаны на идеях кластеризации [1].

Информация о характерных параметрах движения может использоваться, например, при диспетчеризации трафика в заданном районе [2]. Другим перспективным приложением ин-

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-38-90018.

формации о характерных параметрах движения является планирование маршрута перехода судов через акватории с интенсивным движением. Суть проблемы состоит в том, что особенности трафика вдоль предполагаемого маршрута судна оказывают существенное влияние на выбор маршрута из множества возможных. Так, для акваторий с интенсивным движением часто вводится система установления путей движения судов, представляющая собой совокупность ограничений, обусловленных схемой движения судов ("правилами движения"), определенной для конкретной акватории [3]. Указанная схема движения может быть принята как директивно, так и неявно (неформально) как квинтэссенция коллективного опыта судоходства в конкретном районе. При движении судна через такие районы планирование маршрута перехода необходимо осуществлять с учетом накладываемых ограничений.

Классическая задача планирования маршрута перехода представляет собой оптимизационную задачу, целью которой является плавание по самому короткому из возможных путей, за самое короткое время, с наименьшим расходом топлива и т. п. [4, 5]. В основу математической модели рассматриваемого варианта задачи планирования маршрута с учетом ограничений может быть положен поиск кратчайшего пути на взвешенном графе возможных маршрутов судна (такой подход является традиционным и хорошо себя зарекомендовавшим в рассматриваемом классе задач [6]). Вес ребер графа определяется "желательностью" того или иного курса судна для каждой точки акватории с учетом выявленных паттернов движения. В результате выбирается маршрут судна, являющийся наиболее "типичным" (а значит и безопасным) для выбранной акватории.

Рассматриваемая задача планирования маршрутов судов на основе информации о характерном движении представляется актуальной, в том числе, в свете перспективного развития беспилотного судоходства [7, 8].

Основные модельные представления

Введем систему координат x, y , где ось x — географическая долгота, а ось y — географическая широта судна. Определим функцию желательности параметров движения судна $u(x, y, k, v)$, где k — курс, v — скорость, x, y — координаты судна. Желательность движения судна по выбранному маршруту q (некоторая

кривая) может быть выражена как криволинейный интеграл первого рода

$$U = \int_q u(x, y, k, v) dq.$$

Решение оптимизационной задачи выбора маршрута q обеспечивает минимум функционала U :

$$q_* = \arg \min_q U(q).$$

Хотя непосредственное решение последней задачи в принципе возможно, на практике к нему прибегают редко из-за высокой вычислительной сложности и (как правило) дискретности сетки аргументов функции $u(x, y, k, v)$. Популярным эвристическим упрощением задачи является ее модельная интерпретация как задачи поиска кратчайшего пути на взвешенном графе возможных маршрутов судна [6]. Рассмотрим подходы к построению такого графа.

Регулярная сетка вершин. Разбиение акватории на квадратные участки. Разобьем акваторию на множество квадратных участков. На участках, в которых допустимо движение судна, выделим точки центров квадратов. Примем, что эти точки центров квадратов образуют множество вершин графа возможных маршрутов судна. Одна из них соответствует точке начала движения судна, другая — конечной точке маршрута. Множество ребер такого графа может задаваться соединением вершин по принципу "каждая с каждой", исключая те ребра, которые проходят по недопустимым для движения участкам. Также для уменьшения числа ребер графа следует ввести дополнительное условие их максимальной длины. Недостатком такого графа является отсутствие ограничений на величину изменения курса судна в точках вершин, что может порождать неудобные или вообще нереализуемые маршруты.

Регулярная сетка вершин. Слоистое разбиение акватории. Примем, что возможные маршруты судна лежат внутри прямоугольника. Прямоугольник строится так, что начало маршрута лежит в середине одной стороны прямоугольника, а конец маршрута — в середине противоположной стороны; прямоугольник заполняется вершинами графа так, чтобы формировалась правильная прямоугольная сетка. Вершины графа, лежащие на отрезках, перпендикулярных отрезку с вершинами в начальной и конечной точках маршрута, назовем слоями вершин. Примем, что при движении судна воз-

можно переходы только между ближайшими слоями вершин. Ребра, проходящие по недопустимым для движения участкам, исключаются. Также следует ограничить максимальную длину ребра. Такое представление графа учитывает судоводительскую специфику движения — явно ограничивает изменение курса.

Случайная сетка вершин. Возьмем за основу описанное выше разбиение акватории на квадратные участки. Пусть задана вероятность $P(x, y)$ того, что точка центра квадратов с координатами x, y есть вершина графа возможных маршрутов судна. Эта вероятность может быть одинаковой для всей акватории, а может быть различной для разных участков, например, может увеличиваться в местах пересечения судопотоков и на участках характерного изменения курсов. При вырожденном случае $P(x, y) \equiv 1$ множество вершин совпадет с регулярной сеткой. Множество ребер графа задается так же, как для разбиения на квадратные участки. Проблемой случайной сетки вершин является трудность формального обоснования функции $P(x, y)$. Представляется, что она может быть задана лишь эвристическим способом. Кроме того, различные реализации графа возможных маршрутов при его случайном формировании могут приводить к существенно различным результатам поиска наилучшего маршрута. Вместе с тем, после нескольких реализаций возможен последующий выбор "окончательного" наилучшего маршрута по какому-либо критерию.

Сетка вершин на основе ретроспективных данных. Пусть имеются данные о движении судов на выбранной акватории за некоторый период времени в виде множества кортежей значений долготы, широты, скорости и курса судна. Зададим множество вершин графа возможных маршрутов на основе данных этого множества. При этом возможно два основных варианта: множество вершин графа формируется на основе полного множества данных о координатах судов или выбирается только часть множества, как, например, в описанном подходе со случайной сеткой вершин. Множество ребер графа может задаваться так же, как при разбиении на квадратные участки, а может формироваться на основе данных о движении, если известны подмножества кортежей, относящихся к одной траектории. В последнем случае его следует дополнить и "не реализованными" вариантами ребер для обеспечения связности графа. Достоинством такого подхода является формирование возможных маршру-

тов судов, характерных для конкретной акватории. Недостатком является трудность формального обоснования интервала времени, за который берутся данные о движении. Представляется, что выбирать его следует эвристическим способом.

Рассмотрим подходы к заданию веса ребер графа возможных маршрутов. Примем, что в отсутствие данных о движении судов вес ребра равен длине дуги большого круга, соединяющей инцидентные ребру вершины. Если имеются данные о движении судов, вводится весовой коэффициент каждого ребра $a \in [0, 1]$, задаваемый одним из следующих способов.

Учет числа судов с близким курсом и скоростью. Пусть имеются ретроспективные данные о движении судов (множество кортежей значений долготы, широты, скорости и курса) в некоторой окрестности выбранного ребра. Задавая соответствующим ребру курсом и предполагаемой скоростью, подсчитаем число m "близких" (т. е. лежащих в некотором интервале курсов и скоростей) векторов скорости судов в окрестности выбранного ребра. Весовой коэффициент ребра принимается равным $a = 1/m$. Таким образом, ребро будет тем легче (и, соответственно, предпочтительнее при планировании маршрута), чем больше судов ранее двигалось по соответствующей ему траектории. Возможен также подсчет m только на данных о курсе, без учета предполагаемой скорости судна. Недостатком такого подхода является учет всех судов, находившихся в окрестности ребра, без учета их маршрутов. Это может приводить к тому, что планируемый маршрут пройдет по самым "популярным" ребрам, даже если они не являются предпочтительными в конкретном случае.

Учет характерных значений курса и скорости. Для исключения эффекта "популярности" следует определить "характерность" значений курсов и скоростей без явного учета числа судов, их реализующих. Здесь оказывается продуктивной идея кластеризации. Пусть по ретроспективным данным определено множество характерных значений вектора скорости судов, находящихся в окрестности выбранного ребра — кластеров значений вектора скорости. Задавая соответствующим ребру курсом и предполагаемой скоростью, определим, принадлежит ли указанный вектор скорости одному из кластеров: в этом случае весовой коэффициент ребра a задается малым (например, $a = 0,1$). Таким образом, при планировании маршрута предпочтение получают те ребра, ко-

торые соответствуют характерному движению. Также возможно определение кластеров только на основе данных о курсе, без учета предполагаемой скорости судна.

Ребра на основе ретроспективных данных.

Этот способ определения весов ребер может применяться в том случае, если сетка вершин формируется на основе ретроспективных данных (см. выше). Для ребер, сформированных на основе данных о движении, весовой коэффициент ребра задается малым (например, $a = 0,1$). Для тех ребер, которые дополняют исходное множество, делая граф связным, весовой коэффициент принимается большим (например, $a = 1$). Таким образом, предпочтительными становятся уже реализованные маршруты.

При небольшом числе вершин и ребер графа возможных маршрутов судна поиск кратчайшего пути может осуществляться известными детерминированными алгоритмами (Дейкстры, Беллмана—Форда и др.). Сложность наиболее эффективных из них пропорциональна числу ребер и числу вершин (или их логарифму). Если для формирования графа использовать большие выборки ретроспективных данных, число вершин и ребер может оказаться чрезмерно велико для продуктивной работы детерминированных алгоритмов. В этом случае перспективны эвристические алгоритмы: муравьиные и генетические.

Совокупность свойств рассмотренных подходов к построению графа возможных маршрутов судна позволяет предположить, что слоистое разбиение акватории регулярной сеткой вершин является наиболее подходящим для рассматриваемой задачи планирования маршрута через акватории с интенсивным движением. Оно хорошо зарекомендовало себя и в других задачах [9, 10]. Вес ребер задается с учетом характерных значений курса и скорости, определенных путем кластеризации параметров движения.

Для кластеризации вводится метрика — функция, задающая степень близости между объектами [11—13]. В рассматриваемой задаче эту метрику расстояния между объектами 1 и 2 можно ввести следующим образом:

$$D_{12}^2 = w_{lon}(x_1 - x_2)^2 + w_{lat}(y_1 - y_2)^2 + w_{speed}(v_1 - v_2)^2 + w_{course}(k_1 - k)^2.$$

Здесь w_{lon} , w_{lat} , w_{speed} , w_{course} — весовые коэффициенты, которые задаются исходя из данных о характерных размерах кластеров по каждому из измерений. Например, по курсу характерные размеры могут составлять $5...10^\circ$, по скоро-

сти — $2...3$ м/с. Функция разности курсов учитывает периодичность данных по углу. Определение весов по координатам w_{lon} , w_{lat} — это нетривиальная задача, поскольку на акватории могут встречаться как участки маневренного движения размером несколько сотен метров, так и зоны длительного равномерного движения в несколько десятков километров. Поэтому для некоторых приложений (в том числе для рассматриваемой задачи) можно прибегнуть к декомпозиции: осуществлять кластеризацию на отдельных выделенных участках акватории только по скорости и/или курсу.

Для оценки характерных значений курса и скорости представляется подходящими горная и субтрактивная кластеризация, которые не требуют задания числа кластеров [13]. Алгоритм горной кластеризации состоит в следующем. Пусть имеется множество из M объектов и имеется матрица расстояний D_{ij} , задающая степень близости между объектами с индексами i и j . Примем, что сами объекты являются возможными центрами кластеров. Для каждого объекта рассчитывается значение его потенциала:

$$p_i = \sum_{j=1}^M \exp(-\alpha D_{ij}),$$

где α — число, характеризующее масштаб расстояний D_{ij} , $\exp(\cdot)$ — оператор экспоненты. На первом шаге алгоритма выбирают объект с индексом \max_1 , имеющий максимальный потенциал, который и будет центром первого кластера. На втором шаге пересчитывают значения потенциалов объектов по формуле

$$p_i^{(2)} = p_i - p_{\max_1} \exp(-\beta D_{i\max_1}),$$

где β — число, определяющее размер кластеров. Точка с индексом \max_2 с максимальным значением потенциала $p_i^{(2)}$ будет центром второго кластера. Точно так же находятся центры всех следующих кластеров, процедура продолжается, например, пока потенциал очередного кластера превышает некоторый заданный порог или пока имеет место разница между потенциалами соседних уровней.

Субтрактивная кластеризация похожа на горную кластеризацию: после нахождения центра первого кластера из множества объектов исключаются те, которые ему принадлежат. Процедура итерационно повторяется, пока значение потенциала очередного кластера превышает заданный порог.

Результаты расчетов на реальных данных

Исследование проводилось для целого ряда акваторий с использованием реальных данных о движении судов, собранных с ресурса marinetraffic.com с помощью специально разработанной программной системы. Здесь приведен пример успешного решения задачи для Сангарского пролива и Токийского залива. В первом случае для нахождения маршрута судов через акваторию были взяты данные о трафике в течение одной недели (всего около 1,5 млн записей типа "долгота, широта, скорость, курс"), во втором — в течение 3 дней (всего около 2 млн записей).

Граф возможных маршрутов судна формировался на основе слоистого разбиения акватории регулярной сеткой вершин. Параметры разбиения подбирались таким образом, чтобы длина ребер графа не превышала 3 км. Вес ребер задавался с учетом характерных значений курса на отрезке $[0, 360^\circ)$, определенных путем кластеризации данных о движении в окрестности каждого ребра (рис. 1). Если движение по ребру не соответствовало характерному для участка акватории курсу, его вес принимался равным длине дуги большого круга. В противном случае он умножался на весовой коэффициент $a = 0,1$.

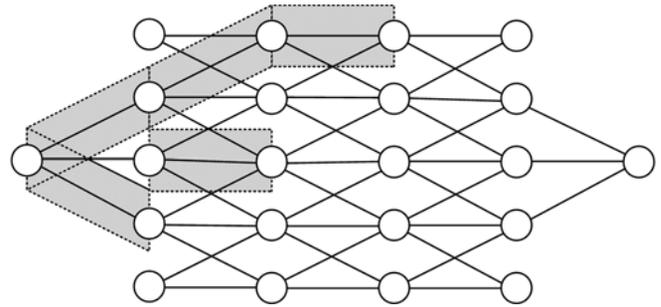


Рис. 1. Граф возможных маршрутов судна. Вес ребер определяется на основе данных о движении в окрестности ребра (выделенные серым области)

Использовался стандартный метод субтрактивной кластеризации. Были приняты следующие параметры субтрактивной кластеризации: параметр α соответствует радиусу кластера 16° ; объекты принадлежат кластеру, если лежат ближе, чем в 20° от его центра ($1,25\alpha$); процедура поиска центров кластеров продолжается, пока потенциал очередного кластера превышает 10 % потенциала первого кластера. Такие значения параметров метода корректно идентифицировали кластеры эталонной выборки данных.

На рис. 2 показаны результаты планирования маршрутов Тихий океан—Японское море (восток—запад) и порт Хакодате—залив Муцу (се-

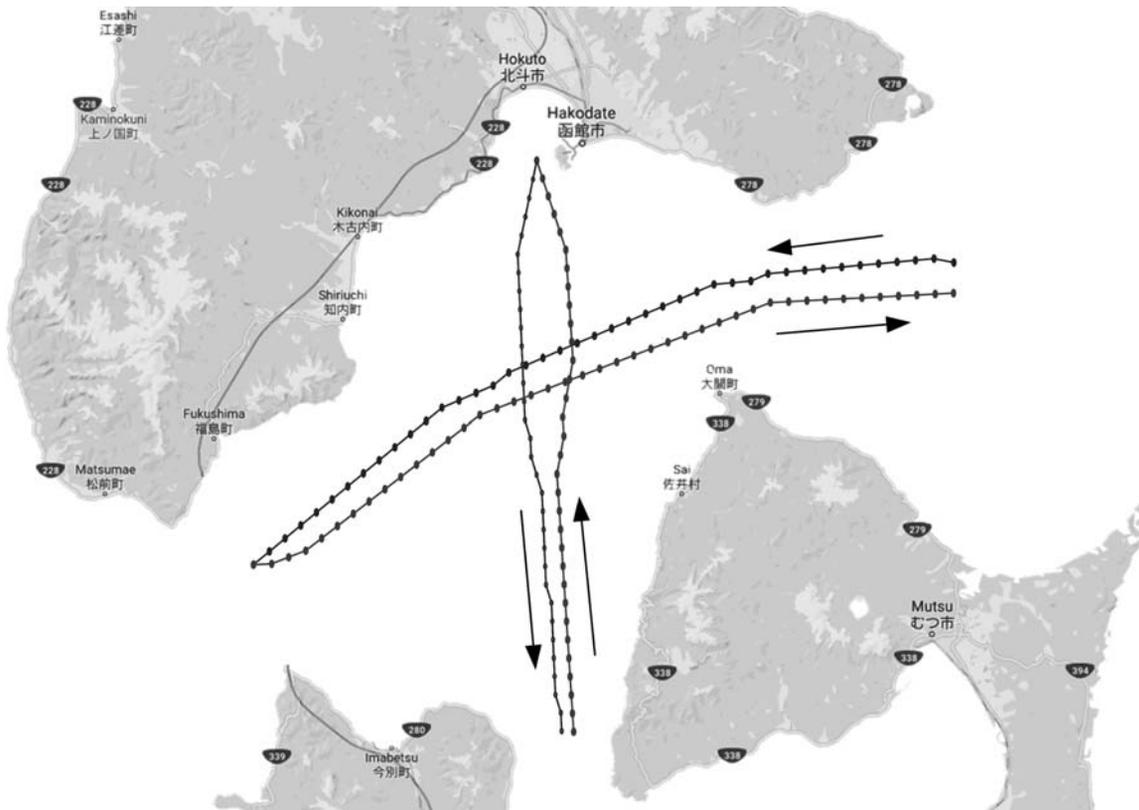


Рис. 2. Результаты планирования маршрута судна в Сангарском проливе



Рис. 3. Результаты планирования маршрута судна в Токийском заливе

вер—юг) и обратно, на рис. 3 — планирование маршрута Тихий океан—Иогама и обратно. Видно, что найденные маршруты полностью соответствуют направлению движения реальных судопотоков, принимают "правильную" сторону движения (показано стрелками, в судовождении принято правостороннее движение).

Обсуждение

В связи с развитием новых информационных технологий и концепции е-навигации наметилась эволюция традиционных моделей задач судовождения в направлении описания коллективного движения судов на акватории в целом. Можно отметить работу [14], в которой на основе классической модели корабельного домена оценивается число опасных сближений судов различных классов на тех или иных участках акватории, выделяются наиболее сложные для движения участки. Этой же задаче посвящены исследования [15—17]. В работе [18] предложен подход для оценки интенсивности движения на акваториях как функции плотности судов.

В исследовании [19] предлагается метод оценки насыщенности трафика, учитывающий геометрические размеры судов. Отмечается, что некоторые акватории обладают ограниченной пропускной способностью для судов больших размеров. Приводится пример, где для безопасного прохождения крупных судов требуется уменьшить их число. В статье [20] предлагается подход к выбору скорости движения судна на основе ретроспективных данных о трафике конкретной акватории (в статье проанализировано движение судов близ Шанхая). Скорость выбирается в зависимости от плотности трафика. Таким образом неявно учитывается коллективный опыт судоводителей.

Подходы, связанные с кластерным анализом данных о движении судов на акватории, также получают все большее развитие. Они исследуют варианты объектов кластеризации, их признаков, метрик и

методов кластеризации. Так, в работе [21] рассматривается задача оценки типичных маршрутов судов на акватории по данным АИС. Метод основан на разбиении акватории на небольшие участки, оценке в них плотности движения, предпочтительных переходов судов между ними, т. е. кластеризация происходит неявно. В результате формируются ломаные — маршруты судов между заданными начальной и конечной точками, учитывающие предыдущий трафик, в том числе принятую схему движения судов. Особенностью (и, по-видимому, существенным недостатком) метода является необходимость последующего сглаживания полученных маршрутов.

В работе [22] кластеризация также используется для оценки типичных для акватории маршрутов. Объектами кластеризации являются ломаные, формируемые по ретроспективным данным АИС. Вводится метрика расстояния между ломаными. Алгоритм кластеризации основан на поиске областей связности, предлагается метод подбора параметров алгоритма. В результате выделяются основные маршруты судов в районе интенсивного судо-

ходства (в работе приведен пример для одного из районов Южно-Китайского моря). Это позволяет выделять аномально движущиеся суда, чей маршрут не характерен для данной акватории. Метод не решает задачу планирования маршрутов. В статье [23] рассматривается похожая задача, объектами кластеризации являются данные о координатах, скоростях и курсах судов. Работа [24] также рассматривает задачу идентификации аномально движущихся судов, признаками являются их координаты, курсы и скорости. Акватория разбивается прямоугольной сеткой, маршруты судов представляются правилами перехода между клетками сетки. Кластеризация основана на построении гистограмм курсов и скоростей для каждой клетки по данным АИС. Аномальные значения скоростей и курсов идентифицируются по этим гистограммам. Хотя предложенный в работе подход может быть положен в основу задачи планирования маршрута судна, она явно не рассматривается. Недостатком метода является необходимость предобработки данных АИС для построения графа маршрутов, высокая вычислительная сложность соответствующих алгоритмов и необходимость большой выборки данных для построения гистограмм.

Настоящая работа также посвящена приложению кластерного анализа данных о движении судов на акватории для решения классической задачи судовождения — планирования маршрута перехода судна. Такой подход обеспечивает новое качество маршрута — его соответствие устоявшимся параметрам движения, выработанным коллективным профессиональным опытом. Постулируя, что ретроспективные данные о движении на акватории являются результатом позитивной эксплуатационной практики, можно предполагать, что спланированный на их основе маршрут будет самым безопасным из возможных в конкретных навигационных условиях.

Упомянутый метод [24], основанный на определении правил движения методами статистики, является наиболее близким к подходу, предложенному в настоящей работе, где предлагается использовать методы кластеризации. Это позволяет достоверно идентифицировать параметры движения и не требует больших массивов исходных данных. Так, в представленных на рис. 2, 3 примерах характерное число данных о курсах судов на "популярных" участках акватории было равно 20...50, на самых насыщенных, как правило, не превышало 200. Это вполне позволило построить устойчивую картину характерных параметров движения.

Вариант графа возможных маршрутов со слоистым разбиением акватории (см. рис. 1) следует признать удачным. На его основе формируются маршруты, не допускающие чрезмерных изменений курса. Характерное число вершин и ребер графа невелико даже для протяженных маршрутов, что позволяет ограничиться детерминированными методами поиска кратчайшего пути.

Заключение

В работе рассматривается задача планирования маршрута перехода судна через акватории с интенсивным движением. Предлагается подход, основанный на использовании идеи, лежащей в основе технологии больших данных: движение судна должно соответствовать характерным кинематическим параметрам движения, определяемым по ретроспективной информации о трафике. Это позволяет учесть коллективный опыт судовождения на конкретной акватории.

Модельные представления задачи включают в себя граф возможных маршрутов судна и функцию "желательности" параметров движения. В работе рассмотрено несколько возможных вариантов модели. Обосновывается выбор в пользу слоистого разбиения акватории регулярной сеткой вершин. Вес ребер графа определяется по результатам кластеризации ретроспективных данных о движении.

Способ построения графа возможных маршрутов судна позволяет уменьшить размерность данных при кластеризации: акватория разбивается на небольшие участки, кластеризация данных о движении осуществляется для каждого из участков отдельно. Признаками объектов являются курсы и скорости движения (совместно или по отдельности). Указывается, что предпочтительнее использовать методы кластеризации, не требующие предварительного задания числа кластеров, например, горной и субтрактивной кластеризации.

Источником данных о трафике акватории могут служить сервисы АИС. В работе показана возможность использования данных АИС, доступных на специализированных интернет-ресурсах. Несмотря на "разреженность" этих выборок данных, они достаточно адекватно представляют сводные особенности трафика.

В работе приведены примеры планирования маршрута перехода через Сангарский пролив и Токийский залив. Видно, что найденные

маршруты достаточно хорошо соответствуют характерному движению судов по акватории, что подтверждает перспективность предложенного подхода для практики.

Список литературы

1. **Zhao L., Shi G.** Maritime anomaly detection using density-based clustering and recurrent neural network // *Journal of Navigation*. 2019. Vol. 72, Iss. 4. P. 894–916.
2. **Гриняк В. М., Девятисильный А. С., Иваненко Ю. С.** Поддержка принятия решений при обеспечении безопасности движения судов на основе кластеризации траекторий // *Вестник государственного университета морского и речного флота им. адмирала С. О. Макарова*. 2020. № 3. С. 436–449.
3. **Общие** положения об установлении путей движения судов, № 9036. М.: Издание ГУНиО МО СССР, 1987. 32 с.
4. **Першина Л. А., Астреина Л. Б.** Выбор маршрута судна на основе погодных условий // *Эксплуатация морского транспорта*. 2019. № 2. С. 30–38.
5. **Wang H. B., Li X. B., Li P. F., Veremey E. I., Sotnikiva M. V.** Application of real-coded genetic algorithm in ship weather routing // *Journal of Navigation*. 2018. N. 4. P. 989–1010.
6. **Сотникова М. В.** Алгоритмы формирования маршрутов движения судов с учетом прогноза погодных условий // *Вестник Санкт-Петербургского университета. Серия 10. Прикладная математика. Информатика. Процессы управления*. 2009. № 2. С. 181–196.
7. **Титов А. В., Баракат Л., Чанчиков В. А., Тактаров Г. А., Ковалев О. П.** Системы управления безэкипажными судами // *Морские интеллектуальные технологии*. 2019. № 1–4 (43). С. 109–120.
8. **Каретников В. В., Козик С. В., Буцанец А. А.** К вопросу оценки рисков использования безэкипажных средств водного транспорта на участке акватории // *Вестник государственного университета морского и речного флота им. адмирала С. О. Макарова*. 2019. № 6. С. 987–1002.
9. **Акмайкин Д. А., Букин О. А., Гриняк В. М., Москаленко М. А.** Планирование маршрута перехода судна с учетом опасности морского волнения // *Морские интеллектуальные технологии*. 2018. № 4–5 (42). С. 148–152.
10. **Акмайкин Д. А., Grnyak V. M.** Ships route searching with respect of sea waves danger // *IOP Conference Series: Earth and Environmental Science*. 2019. Vol. 272. P. 032103.
11. **Bishop C. M.** *Pattern recognition and machine learning*. New York: Springer Science Business Media, 2006. 738 p.
12. **Bezdek J. C.** *Pattern recognition with fuzzy objective function algorithms*. Boston: Springer, 1981. 256 p.
13. **Yager R., Filev D.** *Essentials of fuzzy modeling and control*. New York: John Wiley & Sons, 1994. 408 p.
14. **Debnath A., Chin H. Ch.** Modelling collision potentials in port anchorages: application of the navigational traffic conflict technique (NTCT) // *Journal of Navigation*. 2016. Vol. 69, Iss. 1. P. 183–196.
15. **Гриняк В. М., Иваненко Ю. С., Люлько В. И., Шуленина А. В., Шурыгин А. В.** Цифровое представление и комплексная оценка навигационной безопасности движения на морских акваториях // *Моделирование, оптимизация и информационные технологии*. 2020. № 1. С. 40–41.
16. **Гриняк В. М., Девятисильный А. С., Шуленина А. В.** Оценка эмоциональной нагрузки на судоводителей в условиях коллективного движения // *Вестник государственного университета морского и речного флота им. адмирала С. О. Макарова*. 2019. № 4. С. 640–651.
17. **Гриняк В. М., Горошко О. А., Девятисильный А. С.** Система экспертного оценивания и визуализации параметров траектории безопасного движения судна // *Мехатроника, автоматизация, управление*. 2017. № 2. С. 127–134.
18. **Liu Z., Wu Zh., Zheng Zh.** Modelling ship density using a molecular dynamics approach // *Journal of Navigation*. 2020. Vol. 73, Iss. 3. P. 628–645.
19. **Weng J., Liao Sh., Yang D.** Methodology for estimating waterway traffic capacity at shanghai estuary of the Yangtze river // *Journal of Navigation*. 2020. Vol. 73, Iss. 1. P. 75–91.
20. **Wang L., Li Y., Wan Zh., Yang Z., Wang T., Guan K., Fu L.** Use of AIS data for performance evaluation of ship traffic with speed control // *Ocean Engineering*. 2020. Vol. 204.
21. **Naus K.** Drafting route plan templates for ships on the basis of AIS historical data // *Journal of Navigation*. 2019. Vol. 73, Iss. 3. P. 726–745.
22. **Zhao L., Shi G.** Maritime anomaly detection using density-based clustering and recurrent neural network // *Journal of Navigation*. 2019. Vol. 72, Iss. 4. P. 894–916.
23. **Zhen R., Jin Y., Hu Q., Shao Zh., Niktakos N.** Maritime anomaly detection within coastal waters based on vessel trajectory clustering and naïve Bayes classifier // *Journal of Navigation*. 2017. Vol. 70, Iss. 3. P. 648–670.
24. **Tang H., Wei L., Yin Y., Shen H., Qi Y.** Detection of abnormal vessel behaviour based on probabilistic directed graph model // *Journal of Navigation*. 2019. Vol. 73, Iss. 5. P. 1014–1035.

V. M. Grinyak, Ph.D. (Tech.), Professor,
Vladivostok State University of Economics and Service,
A. V. Shulenina, Senior Lecturer,
Far Eastern Federal University, Vladivostok, Russian Federation

Marine Traffic Data Clustering for Ships Route Planning

This paper is about maritime safety. The system of vessel traffic schemas is one of the key elements of sea traffic control at the areas with heavy traffic. Such system based on a set of rules and guidelines defined by traffic schemas for certain water areas. From the classic approach, vessels that are not following the guidelines do not necessarily create alarming situations at the moment, however, could lead to complex danger navigation situations with the time passed. The problem of ship route planning through the area with highly intensive traffic is considered in this paper. The importance of the problem becomes more significant these days when taking in account development of self-navigating autonomous vessels. It is expected to respect area navigation limitations while planning vessel path through the areas with identified traffic schema. One of the ways to identify navigation limitations could be trajectory pattern recognition at certain sea areas based on retrospective traffic analysis.

Model representation for such task could be based on vessel moving parameters clustering. The presented model is based on solving the shortest path problem on weighted graph. There are several ways to create such weighted graphs are suggested in the paper: regular grid of vertices and edges, layer grid of vertices and edges, random grid of vertices and edges, vertices and edges identified based on retrospective data. All edges are defined as a weighted function of "desirability" of one or another vessel course for each location of sea area with consideration of identified trajectory patterns. For that the area is divided into sub areas where courses and velocity clustering is evaluated. Possible ways of clustering are discussed in the paper and the choice made in favor of subtractive clustering that does not require predefining of cluster count. Automatic Identification Systems (AIS) could be used as data source for the traffic at certain sea areas. The possibility of using AIS data available on specialized public Internet resources is shown in the paper. Although such data typically has low density, they still could well represent vessel traffic features at the certain sea area. In this paper are presenting samples of route panning for Tsugaru Straight ang Tokyo Bay.

Keywords: marine traffic control, route planning, traffic intensity, clustering, big data, graph algorithms, automatic identification system

DOI: 10.17587/it.24.607-615

References

1. **Zhao L., Shi G.** Maritime anomaly detection using density-based clustering and recurrent neural network, *Journal of Navigation*, 2019, vol. 72, iss. 4, pp. 894–916.
2. **Grinyak V. M., Devyatisilnyi A. S., Ivanenko Y. S.** Decision support for marine traffic control based on route clustering, *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova*, 2020, vol. 12, iss. 3, pp. 436–449 (in Russian).
3. **General provisions on the establishment of ship routes**, № 9036, MO SSSR, 1987, 32 p. (in Russian).
4. **Pershina L. A., Astreina L. S.** Ship routing based on weather conditions, *Ekspluatatsiya morskogo transporta*, 2019, iss. 2, pp. 30–38 (in Russian).
5. **Wang H. B., Li X. B., Li P. F., Veremey E. I., Sotnikova M. V.** Application of real-coded genetic algorithm in ship weather routing, *Journal of Navigation*, 2018, iss. 4, pp. 989–1010.
6. **Sotnikova M. V.** Algorithms of marine ships routing taking into account weather forecast, *Vestnik of Saint Petersburg university applied mathematics. Computer science. Control processes*, 2009, iss. 2, pp. 181–196 (in Russian).
7. **Titov A. V., Barakat L., Chanchikov V. A., Taktarov G. A., Kovalev O. P.** Control systems of unmanned vessels, *Marine Intellectual Technologies*, 2019, vol. 43, iss. 1-4, pp. 109–120 (in Russian).
8. **Karetnikov V. V., Kozik S. V., Butsanets A. A.** Risks assessment of applying unmanned means of water transport in the water area, *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova*, 2019, iss. 6, pp. 987–1002 (in Russian).
9. **Akmaykin D. A., Bukin O. A., Grinyak V. M., Moskalenko M. A.** Ships route planing with the account of dangerous sea waves, *Marine Intellectual Technologies*, 2018, vol. 42, iss. 4-5, pp. 148–152 (in Russian).
10. **Akmaykin D. A., Grnyak V. M.** Ships route searching with respect of sea waves danger, *IOP Conference Series: Earth and Environmental Science*, 2019, Vol. 272, p. 032103.
11. **Bishop C. M.** Pattern recognition and machine learning, New York, Springer Science Business Media, 2006, 738 p.
12. **Bezdek J. C.** Pattern recognition with fuzzy objective function algorithms, Boston, Springer, 1981, 256 p.
13. **Yager R., Filev D.** Essentials of fuzzy modeling and control, New York, John Wiley & Sons, 1994, 408 p.
14. **Debnath A., Chin H. Ch.** Modelling collision potentials in port anchorages: application of the navigational traffic conflict technique (NTCT), *Journal of Navigation*, 2016, vol. 69, iss. 1, pp. 183–196.
15. **Grinyak V. M., Ivanenko Y. S., Lulko V. I., Shuleni-na A. V., Shurygin A. V.** Multi-measure navigation safety estimation and digital represent for marine area, *Modelling, Optimization and Information Technologies*, 2020, vol. 8, iss. 1, pp. 40–41 (in Russian).
16. **Grinyak V. M., Devyatisilnyi A. S., Shuleni-na A. V.** Estimation of the emotional load on the navigators under the conditions of heavy traffic, *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova*, 2019, iss. 4, pp. 640–651 (in Russian).
17. **Grinyak V. M., Goroshko O. A., Devyatisilny A. S.** Expert system for evaluation and visualization of the parameters of the marine traffic safety, *Mechatronics, automation, control*, 2017, iss. 2, pp. 127–134 (in Russian).
18. **Liu Z., Wu Zh., Zheng Zh.** Modelling ship density using a molecular dynamics approach, *Journal of Navigation*, 2020, vol. 73, iss. 3, pp. 628–645.
19. **Weng J., Liao Sh., Yang D.** Methodology for estimating waterway traffic capacity at shanghai estuary of the Yangtze river, *Journal of Navigation*, 2020, vol. 73, iss. 1, pp. 75–91.
20. **Wang L., Li Y., Wan Zh., Yang Z., Wang T., Guan K., Fu L.** Use of AIS data for performance evaluation of ship traffic with speed control, *Ocean Engineering*, 2020, vol. 204.
21. **Naus K.** Drafting route plan templates for ships on the basis of AIS historical data, *Journal of Navigation*, 2019, vol. 73, iss. 3, pp. 726–745.
22. **Zhao L., Shi G.** Maritime anomaly detection using density-based clustering and recurrent neural network, *Journal of Navigation*, 2019, vol. 72, iss. 4, pp. 894–916.
23. **Zhen R., Jin Y., Hu Q., Shao Zh., Niktakos N.** Maritime anomaly detection within coastal waters based on vessel trajectory clustering and naïve Bayes classifier, *Journal of Navigation*, 2017, vol. 70, iss. 3, pp. 648–670.
24. **Tang H., Wei L., Yin Y., Shen H., Qi Y.** Detection of abnormal vessel behaviour based on probabilistic directed graph model, *Journal of Navigation*, 2019, vol. 73, iss. 5, pp. 1014–1035.



15—18 марта 2022 г.

**XXIII конференция молодых ученых
"НАВИГАЦИЯ И УПРАВЛЕНИЕ ДВИЖЕНИЕМ"
Санкт-Петербург, Россия**



В формат конференции включаются обзорные лекции ведущих ученых в области теории и практики построения современных систем навигации и управления движением и доклады участников конференции.



31 мая – 21 июня 2022 г. в Санкт-Петербурге
на базе ОАО "Концерн «ЦНИИ «Электроприбор»

**состоится конференции:
XXVIII Санкт-Петербургская
Международная конференция
по интегрированным
навигационным системам**



Тематика конференции

- Инерциальные датчики, системы навигации и ориентации
- Интегрированные системы навигации и управления движением
- Глобальные навигационные спутниковые системы
- Средства гравиметрической поддержки навигации

В рамках каждого направления рассматриваются:

- схемы построения и конструктивные особенности
- методы и алгоритмы
- особенности разработки и применения для различных подвижных объектов и условий движения (аэрокосмические, морские, наземные, подземные)
- испытания и метрология.

Контактная информация:

Тел.: +7 (812) 499 82 10 +7 (812) 499 81 57
Факс: +7 (812) 232 33 76 E-mail: icins@eprib.ru

Адрес редакции:

107076, Москва, Матросская тишина, 23с2

Телефон редакции журнала **+7 916 392 2167**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.
Корректор *М. Ю. Безменова*.

Сдано в набор 10.09.2021. Подписано в печать 25.10.2021. Формат 60×88 1/8. Бумага офсетная.
Усл. печ. л. 8,86. Заказ IT1021. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

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

«ИССЛЕДОВАНИЕ ИЕРАРХИЧЕСКИХ СИСТЕМ НЕЧЕТКОГО ВЫВОДА ПРИ ПОЛУЧЕНИИ ИНТЕГРАЛЬНЫХ ОЦЕНОК АНАЛИЗИРУЕМЫХ ОБЪЕКТОВ»

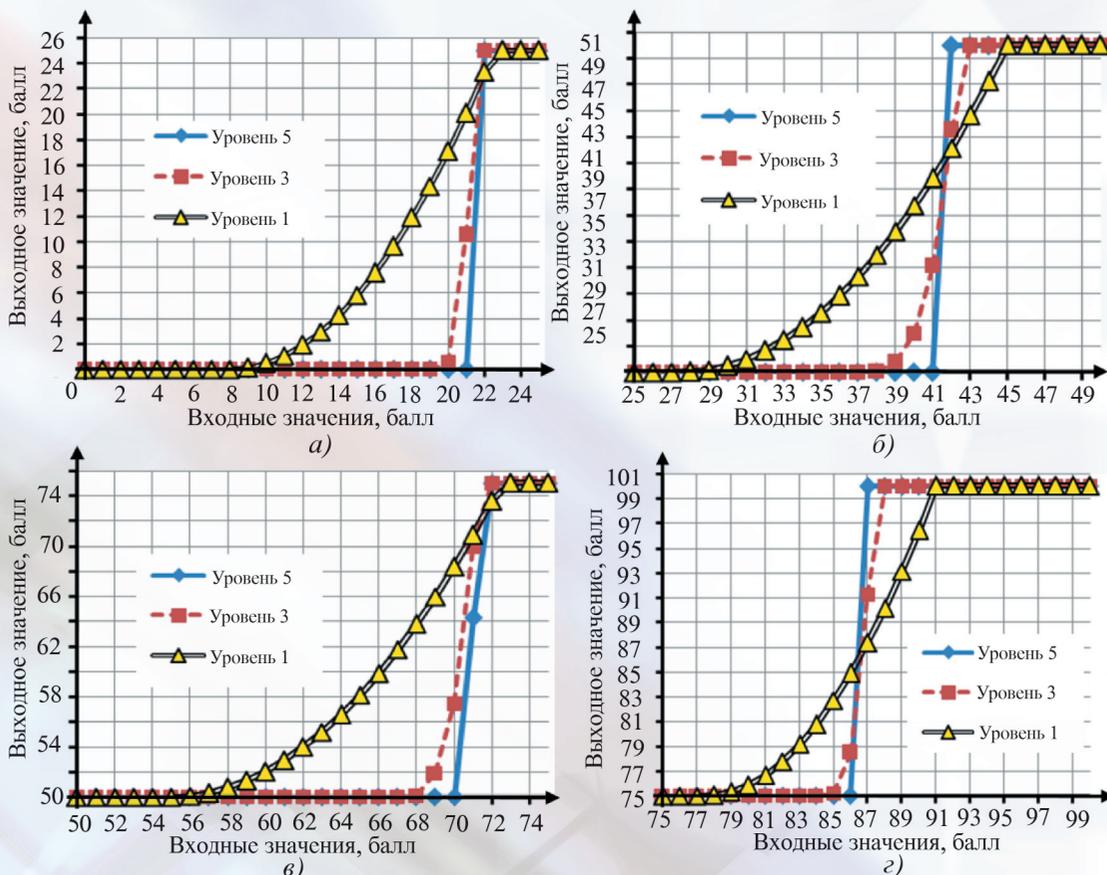


Рис. 8. Результат моделирования ИСНВ-ТС после корректировки термов

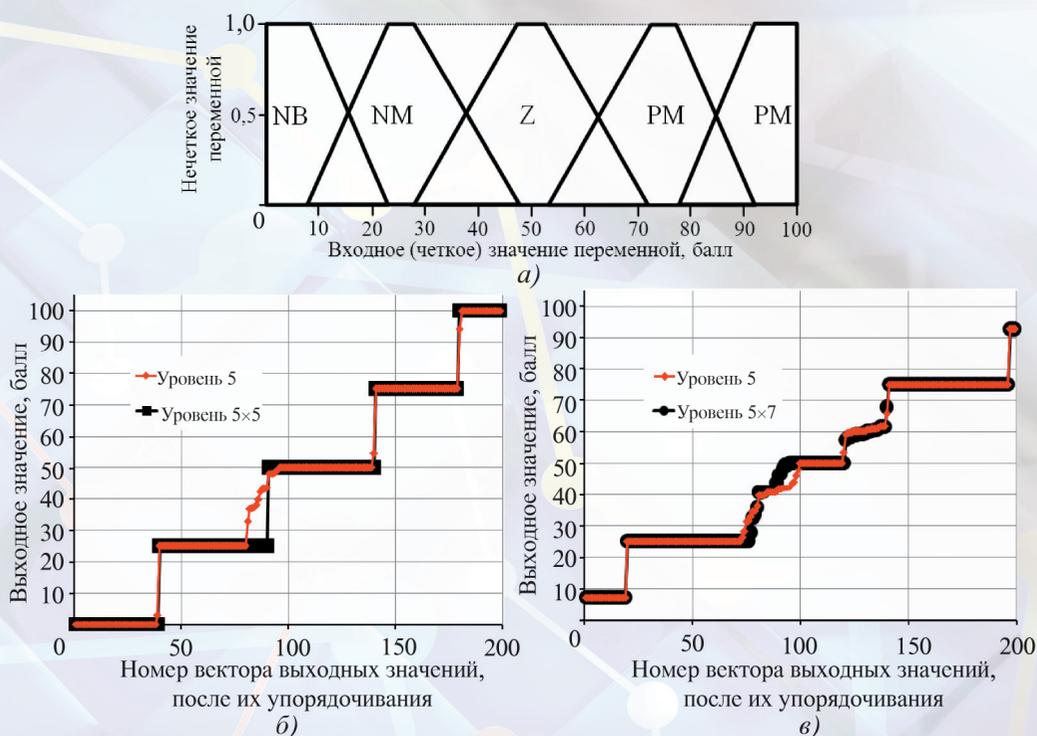


Рис. 9. Результат моделирования ИСНВ-ТС и ИСНВ-М при подаче разнообразных значений

Рисунок к статье В. С. Осиповича, О. С. Медведева, О. Н. Дудич,
В. Л. Красильниковой, К. Д. Яшина

«ТЕХНОЛОГИЯ ПРЕОБРАЗОВАНИЯ РЕЗУЛЬТАТОВ КОМПЬЮТЕРНОЙ ТОМОГРАФИИ В ТРЕХМЕРНЫЕ МОДЕЛИ»

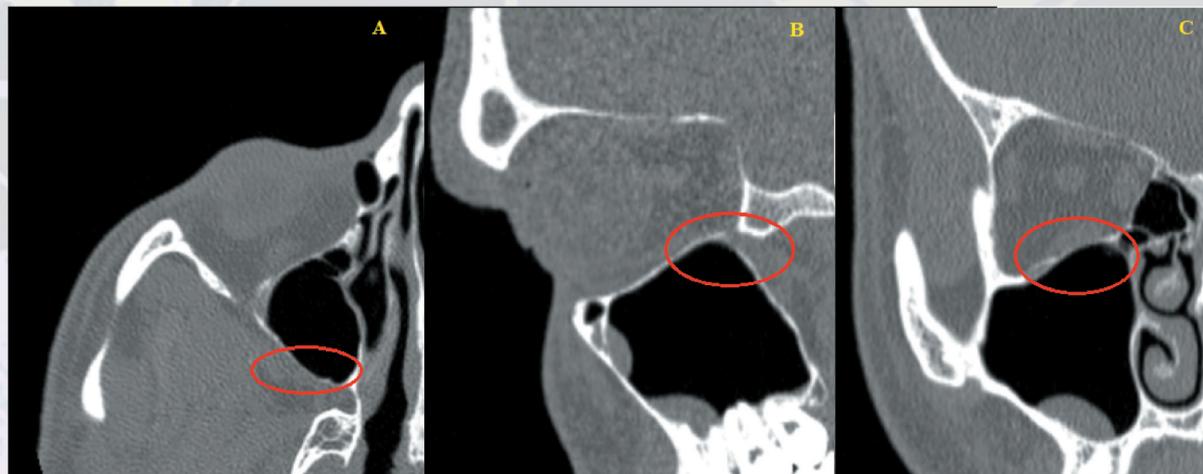


Рис. 1. Изображения трех проекций головы пациента

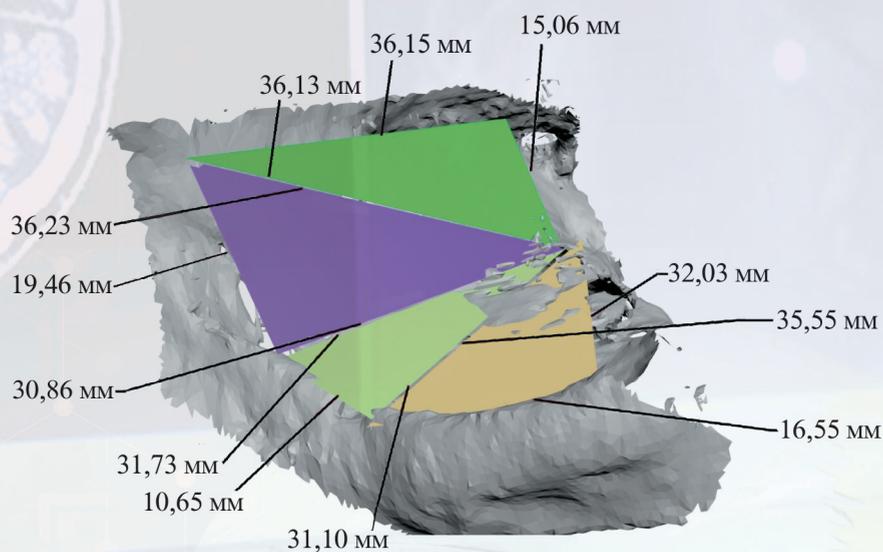


Рис. 3. Размещение треугольников для расчета ориентировочной плоскости

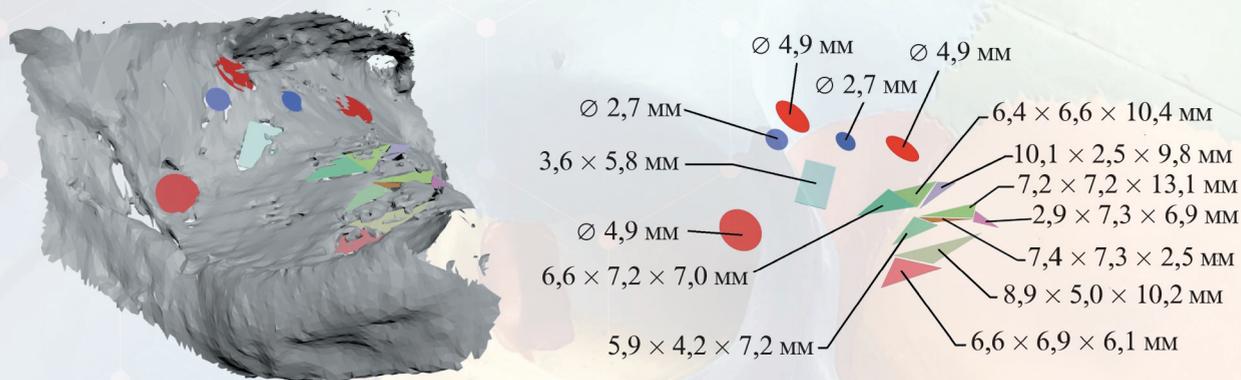


Рис. 4. Круги и прямоугольники, которые вписывались в отверстия трехмерной модели