

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23
2017
№ 4

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Валеева А. Ф., Гончарова Ю. А. Об одном подходе к решению задач операционного планирования по доставке однородной продукции различным клиентам. Часть 2 243
- Деменков Н. П., Микрин Е. А., Мочалов И. А. Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть 1 251
- Вересников Г. С., Панкова Л. А., Пронина В. А. Использование неопределенного программирования в проектировании 258

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Горбаченко В. И., Катков С. Н. Мемристорные клеточные сети для решения крайних задач 266
- Родионов А. Н. Некоторые синтаксические и семантические ограничения сценарных моделей бизнес-процессов 273

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Щеглов К. А., Щеглов А. Ю. Подход к моделированию параметров и характеристик интегрированной информационно-эксплуатационной безопасности . 282
- Авдошин С. М., Лазаренко А. В. Пиринговые ботнеты 290

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ

- Ефимов Ю. С., Матвеев И. А. Выделение точных границ радужки на изображении глаза 300
- Голов А. В., Тимме Е. А., Козлов А. В. Алгоритм автоматизированной оценки параметров работоспособности человека при выполнении нагрузочных тестов 309

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Белим С. В., Ларионов С. Б. Алгоритм выявления поврежденных пикселей изображения на основе выявления сообществ на графах 315

Главный редактор:
СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:
ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:
БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.
СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:
АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕЩУК О. М., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:
БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

CONTENTS

MODELING AND OPTIMIZATION

- Valeeva A. F., Goncharova Ju. A. The Development of the Logistic Transport System 243
- Demenkov N. P., Mikrin E. A., Mochalov I. A. Fuzzy Transformation of Laplace in Tasks of Fuzzy Mathematical Modelling. Part 1 251
- Veresnikov G. S., Pankova L. A., Pronina V. A. Use of Uncertain Programming in Design 258

COMPUTING SYSTEMS AND NETWORKS

- Gorbachenko V. I., Katkov S. N. Memristor Cellular Network for Solving Boundary Value Problems 266
- Rodionov A. N. Some Syntactic and Semantic Limits on Scenary Business Process Models 273

CRYPTOSAFETY INFORMATION

- Shcheglov K. A., Shcheglov A. Yu. Integrated Information-Operational Security Parameters and Characteristics Modeling Approach 282
- Avdoshin S. M., Lazarenko A. V. Peer-to-Peer Botnets 290

INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

- Efimov Yu. S., Matveev I. A. Iris Image Segmentation by Paired Gradient Method with Pupil Boundary Refinement 300
- Golov A. V., Timme E. A., Kozlov A. V. The Algorithm of Automated Estimation of Human Performance Parameters under Load Tests 309

DIGITAL PROCESSING OF SIGNALS AND IMAGES

- Belim S. V., Larionov S. B. The Algorithm of Broken Pixel Detection Based on Community Detection in Graphs 315

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

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

Editors:

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

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

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

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 519.7

А. Ф. Валеева, д-р техн. наук,

профессор каф. вычислительной математики и кибернетики, e-mail: aida_val2004@mail.ru,

Ю. А. Гончарова, аспирант,

Уфимский государственный авиационный технический университет

Об одном подходе к решению задач операционного планирования по доставке однородной продукции различным клиентам. Часть 2*

Приведены результаты численных экспериментов, подтверждающих эффективность разработанных алгоритмов на базе метаэвристики муравьиной колонии, для решения задачи доставки однородной продукции (заказа) различным клиентам. При этом учитываются такие ограничения, как грузоподъемность и неоднородный парк транспортного средства, временные окна, период планирования доставки заказа клиентам, множество депо, отдельная доставка заказа, возможность возврата заказа. В процессе построения маршрутов решалась задача поиска рационального размещения заказа (задача трехмерной упаковки).

Ключевые слова: маршрутизация, размещение заказа в автомобильные транспортные средства, алгоритм муравьиной колонии, основанный на популяции

Введение

Задачи поиска рациональных маршрутов играют важную роль при доставке заказа клиентам точно в срок. Поскольку эти задачи являются *NP*-трудными [5], для их решения широко применяют метаэвристические методы [3, 7]. В работе [10] было установлено, что применение эффективных методов доставки заказа часто дает экономию около 5—20 % от общей его стоимости.

В работе рассматривается задача EVRP о доставке однородного груза в различные регионы России автомобильными транспортными средствами (ТС) различной грузоподъемности, арендуемыми компанией. В одном транспортном средстве находится груз, предназначенный для доставки нескольким клиентам. Компания располагает складом (депо) или несколькими складами для хранения груза, каждое ТС начинает и заканчивает свой маршрут в депо. Компания может установить временной период в размере нескольких дней, в течение которых груз должен быть доставлен клиентам. По пути доставки груза транспортному средству разрешается остановка в некоторые интервалы времени. Кроме того, каждый клиент может быть посещен более чем одним транспортным средством. При этом спрос каждого клиента может быть больше грузоподъемности транспортного средства, и клиент

может быть включен в несколько маршрутов. Масса загружаемого в транспортное средство груза не должна превышать его грузоподъемности. Компания заинтересована в минимизации стоимости пройденного пути. Требуется определить рациональные маршруты доставки однородного груза автомобильными ТС различным клиентам с возможностью одновременного учета рационального размещения груза внутри ТС при построении рациональных маршрутов доставки.

В части 1 статьи были представлены математическая модель задачи EVRP, а также метод P-ACO-EVRP ее решения на базе метаэвристики муравьиной колонии. В части 2 статьи приведены результаты численных экспериментов, подтверждающих эффективность разработанного метода. Приведем математическую модель решаемой задачи.

1. Постановка задачи доставки однородного груза различным клиентам (EVRP)

Приведем математическую модель задачи EVRP. *Исходная информация.*

$G = (V, A)$ — граф с множеством вершин V и множеством дуг A ;

V — множество вершин, которое делится на два подмножества $V_c = \{1, \dots, n\}$ и $V_h = \{n+1, \dots, n+k\}$;

$V_c = \{1, \dots, n\}$ — множество городов и клиентов;

$V_h = \{n+1, \dots, n+k\}$ — множество депо;

* Часть 1 статьи опубликована в 2016 г., том 22, № 10.

$A^{NAtype} = \{A_1^{NAtype}, \dots, A_{|A|}^{NAtype}\}$ — множество ребер графа — дороги с заданными ограничениями;
 $NAtype$ — число ограничений;
 m_u — число ТС, расположенных в каждом депо $u \in V_h$, $m_u \in \{1, \dots, r_u\}$;
 r_u — максимальное число ТС, расположенных в депо $u \in V_h$;
 Q_v — грузоподъемность ТС v из депо $u \in V_h$, $Q_v \in Z^+$, где $v \in [\underline{v}, \bar{v}]$. Предполагается, что $\underline{v} = 1$ и $\bar{v} = r_u$;
 q_i — объем спроса в грузе, связанный с каждым клиентом $i \in V_c$;
 p_i — объем возвращаемого груза, связанный с каждым клиентом $i \in V_c$;

$cost_v$ — стоимость пройденной единицы расстояния ТС v , которая включает в себя стоимость аренды ТС v на единицу расстояния $cost_{rent}^v$, затраты на бензин ТС v на единицу расстояния $cost_{petrol}^v$:
 $cost_v = cost_{rent}^v + cost_{petrol}^v$;

$[a_i, b_i]$ — временное окно для клиента $i \in V_c$;

M дней — период планирования доставки груза.

Введем ряд обозначений:

c_{ij} — расстояние между пунктами i и j , $i, j \in V_c$;

e_{ij} — время в пути между пунктами i и j , $i, j \in V_c$;

s_i — время обслуживания клиента $i \in V_c$;

$penalty_time_i$ — штрафная стоимость за начало обслуживания клиента $i \in V_c$ после b_i , пропорциональная времени опоздания ТС;

w_i^v — время начала обслуживания клиента $i \in V_c$ ТС v из депо $u \in V_h$;

x_{ij}^v — переменная логического типа, принимающая значение 1, если ТС v из депо $u \in V_h$ перемещается в направлении от пункта i к пункту j , и значение 0 в противном случае, $i, j \in V_c$;

y_i^v — это спрос i -го клиента, обслуженного ТС v из депо $u \in V_h$;

d_i^v — это объем требования i -го клиента на возврат груза, осуществленный ТС v из депо $u \in V_h$;

$penalty_pack_v$ — штрафная стоимость за маршрут, не соответствующий рациональному размещению груза в ТС v из депо $u \in V_h$, т. е. такому, при котором очередной груз не умещается в ТС v .

Преобразуем исходный граф $G = (V, A)$ в граф $G' = (V', A')$ с учетом ограничений $\{A^1, \dots, A^{NAtype}\}$, в котором $A' = A \cup A^1 \cup A^2 \cup \dots \cup A^{NAtype}$, $V' = V$.

В графе $G' = (V', A')$ требуется минимизировать стоимость пройденного пути:

$$\left(\sum_{v=1}^{m_u} cost_v \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^v + \sum_{v=1}^{m_u} \sum_{i=1}^n \sum_{j=1}^n penalty_time_j x_{ij}^v \right) \rightarrow \min; \quad (1)$$

$$u = n + 1, \dots, n + k;$$

$$c_{ij} \in A' = A \cup A^1 \cup A^2 \cup \dots \cup A^{NAtype};$$

при выполнении следующих условий: пройденный путь включает каждую вершину не менее одного раза

$$\sum_{v=1}^{m_u} \sum_{i=1}^n x_{ij}^v \geq 1;$$

$$j = 1, \dots, n; u = n + 1, \dots, n + k; \quad (2)$$

$$\sum_{v=1}^{m_u} \sum_{i=1}^n x_{iz}^v - \sum_{v=1}^{m_u} \sum_{j=1}^n x_{zj}^v = 0;$$

$$z = 0, \dots, n; u = n + 1, \dots, n + k; \quad (3)$$

спрос каждого клиента i в грузе должен быть удовлетворен только при условии, что ТС v проходит через клиента i :

$$y_i^v \leq \sum_{j=1}^n x_{ij}^v q_j;$$

$$i = 1, \dots, n; v = 1, \dots, m_u; u = n + 1, \dots, n + k; \quad (4)$$

объем требования каждого клиента i на возврат груза должен быть удовлетворен только при условии, что ТС v проходит через клиента i :

$$d_i^v \leq \sum_{j=1}^n x_{ij}^v p_j;$$

$$i = 1, \dots, n; v = 1, \dots, m_u; u = n + 1, \dots, n + k; \quad (5)$$

количество груза, размещенного в ТС v , не должно превышать грузоподъемности ТС v :

$$\sum_{i=1}^n (y_i^v - d_i^v) \leq Q_v;$$

$$v = 1, \dots, m_u; u = n + 1, \dots, n + k; \quad (6)$$

учитываются временные окна:

$$a_i \leq w_i^v \leq b_i;$$

$$i = 1, \dots, n; v = 1, \dots, m_u; u = n + 1, \dots, n + k; \quad (7)$$

$$\sum_{j=1}^n x_{ij}^v w_j^v \geq \sum_{i=1}^n x_{ji}^v (w_i^v + s_i + e_{ij});$$

$\forall i, j \in \{1, \dots, n\}; v = 1, \dots, m_u; u = n + 1, \dots, n + k; (8)$

и M — дневное планирование доставки груза:

$$\sum_{v=1}^{m_u} \sum_{i=1}^n \sum_{j=1}^n x_{ij}^v (e_{ij} + s_j) \leq M; u = n + 1, \dots, n + k; (9)$$

груз должен умещаться в ТС v :

$$fpack_v = \begin{cases} \leq 1, & penalty_pack_v = 0 \\ > 1, & penalty_pack_v = \text{const} > 0; \end{cases}$$

$$v = 1, \dots, m_u. \quad (10)$$

Ограничение (10) в рассматриваемой задаче доставки однородного груза различным клиентам с

учетом ряда ограничений является необязательным и дает возможность учитывать рациональное размещение груза внутри ТС, если это необходимо.

Для решения (1)–(10) разработан метод P-ACO-EVRP на основе метаэвристики муравьиной колонии, основанной на популяции. Этот метод описан в работе [1]. Приведем основные шаги алгоритма P-ACO-EVRP.

Шаг 1. МЕТОД КИНИ-РАЙФА (из исходного графа маршрутов $G = (V, A)$ формируется матрица расстояний c_{ij} , из которой выбираются дуги, удовлетворяющие взаимонезависимым по предпочтениям условиям $\{A^1, \dots, A^{N_{Аtype}}\}$ (например, качество дорог, ограничения скорости на дорогах и т.д.), строится новый граф $G' = (V', A')$, из которого формируется матрица расстояний c'_{ij} , ищется кратчайший путь по алгоритму муравьиной колонии, основанному на популяции (*Population Based Ant Colony Optimization, P-ACO*).

Шаг 2. МЕТОД БЛИЖАЙШЕГО СОСЕДА (процедура распределения клиентов по депо).

Шаг 3. АЛГОРИТМ ЗАМЕТАНИЯ (процедура распределения клиентов по транспортным средствам).

Шаг 4. Процедура поиска рациональных маршрутов.

Шаг 4.1. ПОИСК РАЦИОНАЛЬНЫХ МАРШРУТОВ.

Шаг 4.2. РАЗМЕЩЕНИЕ ЗАКАЗА.

Шаг 4.3. ОБНОВЛЕНИЕ ПОПУЛЯЦИИ ЛУЧШИХ МАРШРУТОВ (стратегии обновления популяции *Age, Quality, Prob* и *Elitism*).

Шаг 4.4. ВЫЧИСЛЕНИЕ МАТРИЦЫ ПОПУЛЯЦИИ.

Шаг 5. Улучшающая процедура 2_OPT_INTERCHANGE (обмен вершинами между маршрутами).

Для исследования разработанного алгоритма проводили численные эксперименты для различных тестовых наборов данных.

2. Численные эксперименты

Для исследования эффективности предложенного алгоритма P-ACO-EVRP были проведены численные эксперименты. Поскольку математическая модель задачи EVRP содержит шесть основных ограничений задач маршрутизации ТС, а именно: грузоподъемность транспортного средства, временные окна, период планирования, множество депо, раздельная доставка, возможность возврата груза, эффективность предложенного алгоритма P-ACO-EVRP была исследована на тестовых наборах с наилучшими известными решениями для математических моделей CVRP, VRPTW, MDVRP, MDVRPTW, SDVRP, SDVRPTW, PVRP, PVRPTW, VRPPD. При этом были сгенерированы тестовые наборы данных для решаемой задачи EVRP, учитывающей, кроме перечисленных выше ограничений, рациональное размещение груза внутри ТС во время построения рациональных маршрутов.

Эксперимент 1. Влияние параметров на качество работы алгоритма P-ACO-EVRP

Цель данного эксперимента — определение лучших параметров алгоритма: β (привлекательность перехода из одного города в другой), α (количество феромона). Были выбраны три тестовых набора R101, C101, RC108 из 56 тестовых наборов, предложенных в работе [9]. Для каждого тестового набора рассматривалось три депо. Учитывались условия раздельной доставки, периода планирования, возврата товаров для соответствующих тестовых наборов [4, 6, 8].

Эксперимент 1.1. Исследовалось влияние параметра α (параметр значимости количества феромона при выборе следующего клиента). Значения параметра α менялось от 1 до 10; $\beta = 1$; $\rho = 0,2$.

Результаты в табл. 1 показывают, что увеличение параметра α приводит к существенному ухудшению решений.

Эксперимент 1.2. Исследовалось влияние параметра β (параметр значимости расстояния при выборе следующего клиента). Значение параметра β менялось от 1 до 10, $\alpha = 1$, $\rho = 0,2$.

Результаты в табл. 2 показывают, что изменение значения β практически не влияет на решение, лучшее решение достигается при значениях параметра, равных 3 и 5.

Эксперимент 1.3. В табл. 3 приведены некоторые результаты для различных комбинаций α и β .

Эксперимент 1.4. В табл. 4 приведены значения лучших решений, полученных для трех тестовых задач r4R101, r6C101, r4RC108.

При увеличении параметра α решение задачи ухудшается, изменение β , в свою очередь, подобного эффекта не дает.

Таблица 1

Влияние параметра α на качество решения L
(L — длина маршрута в усл. ед.)

α	1	3	5	7	9
L	4225,73	4615,072	5536,71	5453,55	5487,92

Таблица 2

Влияние параметра β на качество решения L
(L — длина маршрута в усл. ед.)

β	1	3	5	7	9
L	4363,47	4195,52	4308,29	4309,89	4392,26

Таблица 3

Влияние комбинаций параметров α и β на качество решения L
(L — длина маршрута в усл. ед.)

α	β	L	α	β	L
2	3	4223,83	2	12	4300,29
2	4	4489,04	3	5	4485,82
2	5	4498,62	3	10	4195,52
2	8	4279,48	3	12	4501,02
2	10	4237,10	3	15	4509,82

Таблица 4

Лучшие полученные решения (L — длина маршрута в усл. ед.)

Задача	Лучшее полученное решение L
p4R101	4195,52
p6C101	4179,52
p4RC108	3060,81

Выводы по эксперименту: результаты численных экспериментов показали, что комбинации параметров α и β оказывают существенное влияние на качество получаемого решения. Наилучшие результаты достигаются при значениях β в 3–5 раз больших, чем α .

Эксперимент 2. Определение влияния стратегий обновления популяции алгоритма P-ACO-EVRP на качество решения

Цель данного эксперимента заключается в определении влияния стратегий обновления популяции *Age*, *Quality*, *Prob* и *Elitism* алгоритма P-ACO-EVRP на качество решения задачи EVRP. При этом проводилось сравнение алгоритма P-ACO-EVRP с муравьиным алгоритмом *Ant Colony System (ACS)* [2].

Для численного эксперимента были выбраны тестовые наборы данных R101, C101, RC108, R201, C208, RC201 из 56 тестовых наборов, предложенных в работе [9]. Для каждого тестового набора рассматривали три депо, условия раздельной доставки, периоды планирования, возврата товаров (данные были взяты из известных работ для соответствующих тестовых наборов) [4, 6, 8].

Выводы по эксперименту: из результатов, представленных в табл. 5, где даны значения длины маршрута в условных единицах, следует, что решения, полученные алгоритмом P-ACO-EVRP с разными стратегиями обновления популяции отличаются по качеству решения между собой не более, чем на 2%. Стратегия *Prob* показала лучшие результаты по качеству решения по сравнению со стратегиями *Quality*, *Age*, *Elitism*. Стратегия *Quality* удалось получить лучшее решение на 1-м тестовом

Таблица 5

Сравнение результатов работы алгоритма P-ACO-EVRP с разными стратегиями обновления популяции для задачи EVRP

Номер примера	L , усл. ед.				
	Стратегия <i>Age</i>	Стратегия <i>Quality</i>	Стратегия <i>Prob</i>	Стратегия <i>Elitism</i>	Алгоритм <i>ACS</i>
p4R101	4207,07	4198,09	4195,52	4209,36	4223,53
p6C101	4200,23	4179,52	4185,80	4188,30	4224,16
p4RC108	3070,91	3063,77	3060,81	3062,68	3084,36
p6R201	5927,11	5910,75	5904,67	5921,77	5947,82
p4C208	1620,97	1604,72	1597,20	1607,49	1614,76
p6RC201	5865,91	5935,29	5834,97	5952,17	5891,52

наборе p6C101. Алгоритму ACS удалось получить результат лучший по сравнению с алгоритмом P-ACO-EVRP со стратегией *Quality* лишь на одном тестовом наборе p6RC201. Все остальные решения, полученные алгоритмом ACS, хуже решений, полученных алгоритмом P-ACO-EVRP с разными стратегиями обновления популяции.

3. Анализ эффективности работы программного обеспечения для решения прикладных задач

Для оценки эффективности предлагаемого подхода для решения задачи доставки однородного груза различным клиентам рассматривается пример с использованием реальных данных нефтехимического предприятия по критерию минимизации стоимости пройденного пути. Решение, полученное с использованием предлагаемого метода, сравнивается с решением, полученным известным ПО logvpr [11] (*route optimization and fleet planning and scheduling web application and service*).

Таблица 6

Типы используемых емкостей параллелепипедной формы и их характеристики

Тип емкости	Масса, кг	Длина, см	Ширина, см	Высота, см
I	727	120	80	100
II	723	120	80	100
III	713	120	80	100
IV	1178	120	100	116
V	1172	120	100	116
VI	1176	120	100	116
VII	1190	120	100	116
VIII	21,95	28,5	22	41
IX	23,625	31,7	28,3	32
X	28,961	31,7	28,3	37,5
XI	34,125	32,5	28,3	43,3

Таблица 7

Типы используемых емкостей цилиндрической формы и их характеристики

Тип емкости	Масса, кг	Высота, см	Диаметр, см
XII	240,7	88,2	59,5
XIII	246,625	88,2	59,5
XIV	71,65	62,5	40
XV	55	60	38
XVI	71,65	65	40
XVII	131,55	80	45
XVIII	239,55	100	55
XIX	239,55	101	60
XX	247,95	97,8	58,1
XXI	246,75	93,5	58,1
XXII	230	84,5	59,4
XXIII	57	56,6	37,1
XXIV	57,1	56	38,6
XXV	113,9	68,6	44
XXVI	113,9	68,6	44

Таблица 8

Характеристики ТС

Тип ТС	Грузоподъемность, т	Длина, м	Ширина, м	Высота, м	Стоимость 1 часа аренды, руб.
1	20	14	2,4	2,35	1200
2	10	8	2,3	1,8	820
3	7	6	2,35	2,35	820

Таблица 9

Характеристики депо предприятия

Депо	Число ТС				Период планирования, дни
	1	2	3	4	
Уфа	5	—	5	—	—
Москва	10	5	10	5	—
Владивосток	7	5	7	6	7

Предприятие производит нефтехимическую продукцию — реагент. Емкости с реагентом хранятся на трех складах (депо), количество реагента на складах различно. Каждое депо имеет свой парк автомобильных транспортных средств разной грузоподъемности, арендуемых предприятием.

Необходимо доставить емкости с реагентом из Уфы, Москвы и Владивостока в следующие города: Октябрьский, Лениногорск, Альметьевск, Набережные Челны, Нефтекамск, Сарапул, Чернушка, Самара, Саратов, Нижний Новгород, Казань, Воронеж, Волгоград, Сызрань, Находка, Хабаровск, Комсомольск-на-Амуре, Улан-Удэ, Иркутск.

Выше приведены таблицы с рассматриваемыми видами емкостей и их характеристиками (табл. 6, 7).

В табл. 8 представлены характеристики имеющихся ТС, в табл. 9 — характеристики депо предприятия.

В табл. 10 представлено распределение емкостей на доставку по городам потребителей, в табл. 11 — распределение емкостей на возврат по городам потребителей.

В табл. 12 показаны временные окна и время обслуживания для тех клиентов, для которых заданы эти данные.

Приведем результаты работы прототипа разработанного ПО на базе алгоритма P-ACO-EVRP и результаты работы ПО **logvnp**. Стоимость 1 л дизельного топлива примем равной 36 руб.

Таблица 10

Распределение емкостей на доставку по городам потребителей

Город	Тип емкости	Число емкостей (доставка)
Октябрьский	IV	8
	III	2
	X	33
Лениногорск	VIII	15
	IX	20
	X	10
Альметьевск	III	2
	IX	5
	XI	20
Набережные Челны	I	3
	IX	25
	XI	24
Нефтекамск	IX	1
	XIII	5
	XVIII	9
Сарапул	XII	6
	XIII	3
	XIX	7
Чернушки	XXI	9
	XII	2
	XVII	6
Самара	XXII	8
	X	20
	VIII	20
Саратов	IV	2
	XI	30
	IX	50
Нижний Новгород	IX	30
	X	30
	XI	45
Казань	IX	20
	X	10
	IX	20
Воронеж	IX	20
	IX	40
	IX	40
Волгоград	IX	30
	X	30
	IX	40
Сызрань	X	50
	XI	100
	IX	40
Находка	X	50
	XI	100
	IX	40
Хабаровск	IX	30
	X	30
	IX	40
Комсомольск-на-Амуре	X	50
	XI	100
	IX	40
Улан-Удэ	IX	40
	IX	55
	X	40
Иркутск	IX	55
	X	40
	X	40

Таблица 11

Распределение емкостей на возврат по городам потребителей

Город	Тип емкости	Число емкостей (возврат)
Октябрьский	IV	—
	III	—
	X	—
Лениногорск	VIII	—
	IX	—
	X	—
Альметьевск	III	—
	IX	—
	XI	—
Набережные Челны	I	—
	IX	—
	XI	—
Нефтекамск	IX	—
	XIII	—
	XVIII	—
Сарапул	XII	—
	XIII	—
	XIX	—
Чернушки	XXI	—
	XII	—
	XVII	—
Самара	XXII	—
	X	—
	VIII	—
Саратов	IV	—
	XI	—
	IX	—
Нижний Новгород	IX	—
	IX	—
	X	—
Казань	IX	—
	X	—
	XI	—
Воронеж	IX	—
	X	—
	IX	—
Волгоград	IX	—
	IX	—
	IX	10
Сызрань	IX	—
	X	10
	IX	7
Находка	X	12
	XI	5
	IX	—
Хабаровск	IX	—
	X	10
	IX	7
Комсомольск-на-Амуре	X	12
	XI	5
	IX	—
Улан-Удэ	IX	—
	IX	15
	X	10
Иркутск	IX	15
	X	10
	X	10

Таблица 12

Временные окна и время обслуживания клиентов

Город	Начало временного окна	Конец временного окна	Время обслуживания, мин
Самара	8:00	9:00	60
Саратов	17:00	19:00	70
Нижний Новгород	10:30	11:30	60
Казань	16:30	18:30	120
Воронеж	14:00	16:30	60
Волгоград	9:00	10:00	60
Сызрань	16:00	19:00	60

**1. Депо Уфа.
P-ACO-EVRP**

Маршрут 1 (используется ТС типа 1) (рис. 1):
Уфа—Октябрьский—Лениногорск—Альметьевск—Набережные Челны—Уфа.

Длина маршрута: 712 км, грузоподъемность ТС: 20 000 кг, загруженность ТС: 17987,323 кг, стоимость маршрута (бензин + аренда ТС): 17 421 руб.

План упаковки однородного груза в ТС типа 1 в маршруте 1 представлен на рис. 2 (см. четвертую сторону обложки).

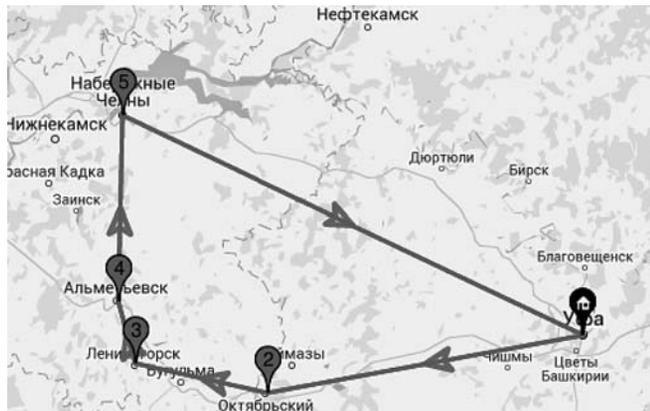


Рис. 1. Маршрут 1 на карте (ПО P-ACO-EVRP)

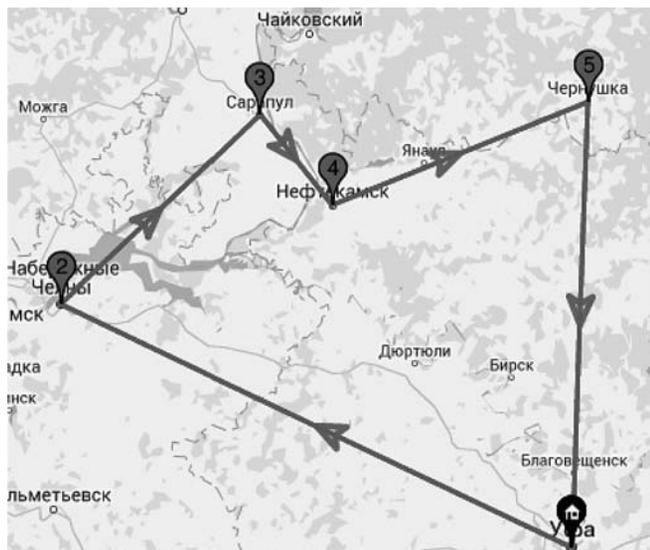


Рис. 3. Маршрут 2 на карте (ПО P-ACO-EVRP)



Рис. 5. Маршрут 1 на карте (ПО logvrp)

Единиц груза всего: 166 шт.

Единиц груза упаковано: 166 шт. (100 %).

Из них:

- емкость (тип 1) — 2 шт.;
- емкость (тип 10) — 43 шт.;
- емкость (тип 11) — 44 шт.;
- емкость (тип 3) — 4 шт.;
- емкость (тип 4) — 8 шт.;
- емкость (тип 8) — 15 шт.;
- емкость (тип 9) — 50 шт.

Объем груза: 21,92 м³ (66 % объема).

Масса груза: 17 987,323 кг (90 % грузоподъемности).

Маршрут 2 (используется ТС типа 1) (рис. 3):
Уфа—Чернушка—Сарапул—Нефтекамск—Набережные Челны—Уфа.

Длина маршрута 1009 км, грузоподъемность ТС 20 000 кг, загруженность ТС 13 332,075 кг, стоимость маршрута (бензин + аренда ТС) 31 078 руб.

План упаковки однородного груза в ТС типа 1 в маршруте 2 представлен на рис. 4 (см. четвертую сторону обложки).

Единиц груза всего: 57 шт.

Единиц груза упаковано: 57 шт. (100 %).

Из них:

- емкость (тип 1) — 1 шт.;
- емкость (тип 12) — 8 шт.;
- емкость (тип 13) — 8 шт.;
- емкость (тип 17) — 6 шт.;
- емкость (тип 18) — 9 шт.;
- емкость (тип 19) — 7 шт.;
- емкость (тип 21) — 9 шт.;
- емкость (тип 22) — 8 шт.;
- емкость (тип 9) — 1 шт.

Объем груза: 17,45 м³ (22 % объема).

Масса груза: 13 332,075 кг (67 % грузоподъемности).

Общая стоимость маршрута: 48 499 руб.

В городе Набережные Челны возникла отдельная доставка, связанная с тем, что все емкости для клиентов из маршрута 1 не уместились в ТС типа 1, поскольку алгоритм учитывает рациональное размещение груза внутри ТС при построении рациональных маршрутов доставки.

Задачи, решаемые с помощью ПО **logvrp**, не учитывают отдельную доставку и рациональное размещение груза внутри ТС при построении рациональных маршрутов доставки. Подобные условия не учитываются при решении подобных задач и в других пакетах (ArcGIS, IBM ILOG, Деловая карта, ТопПлан).

При расчете маршрутов с помощью ПО **logvrp** будем использовать те же ТС, с которыми были построены маршруты 1 и 2, в качестве депо взят город Уфа.

Маршрут 1 (используется ТС типа 1) (рис. 5):

Уфа—Октябрьский—Лениногорск—Альметьевск—Набережные Челны—Уфа—Набережные Челны—Уфа.

Длина маршрута 1293 км.

Стоимость маршрута (бензин + аренда ТС) 31 636 руб.

В городе Набережные Челны все емкости для клиента не уместились в ТС типа 1, поэтому ТС типа 1 пришлось вернуться в депо в город Уфа, сделать дозагрузку и вернуться обратно к клиенту в городе Набережные Челны, за счет чего увеличилась стоимость пройденного ТС типа 1 пути на маршруте 1.

Маршрут 2 (используется ТС типа 1) (рис. 6):

Уфа—Чернушка—Сарапул—Нефтекамск—Уфа.

Длина маршрута: 746 км, загруженность ТС 12 605,075 кг, стоимость маршрута (бензин + аренда ТС) 22 977 руб.

Общая стоимость маршрута: 54 613 руб.

Таким образом, общая стоимость маршрута (54 613 руб.), полученная с помощью ПО **logvpr** больше стоимости маршрута, полученной алгоритмом P-ACO-EVRP (48 499 руб.) на 12,6 %. Это объясняется тем, что в пакете **logvpr** не учитывается ограничение раздельной доставки, а также рациональное размещение груза внутри ТС при построении рациональных маршрутов доставки.

2. Депо Москва.

Маршрут 3 (используется ТС типа 3):

Москва—Воронеж—Самара—Саратов—Волгоград—Москва.

Длина маршрута 3234 км.

Загруженность ТС 5912,58 кг.

Стоимость маршрута (бензин + аренда ТС) 79 126 руб.

Маршрут 4 (используется ТС типа 3):

Москва—Нижний Новгород—Казань—Сызрань—Москва.

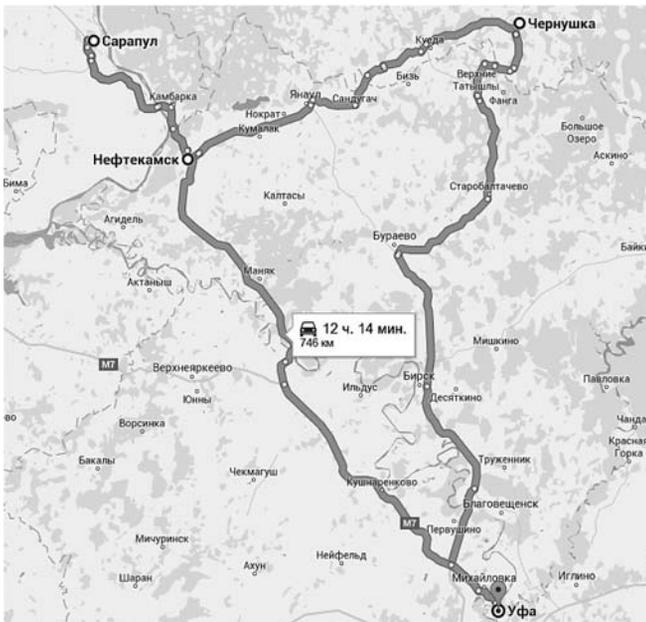


Рис. 6. Маршрут 2 на карте (ПО logvpr)

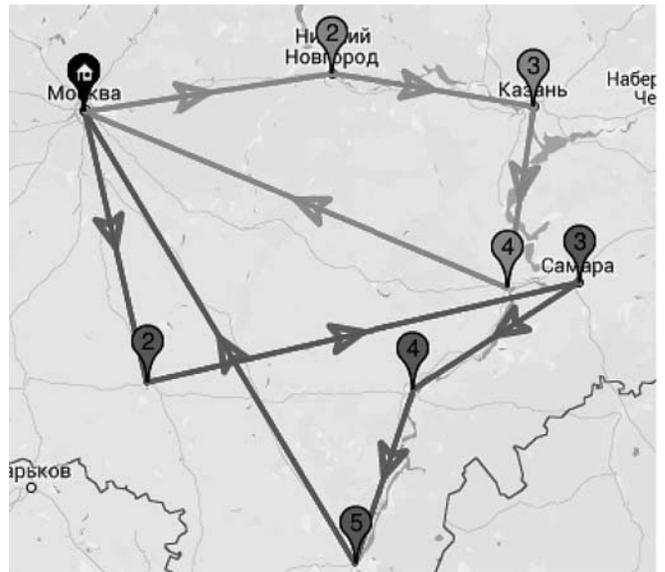


Рис. 7. Маршрут двух ТС типа 3 из депо Москва

Длина маршрута 2065 км.

Загруженность ТС 5239,455 кг.

Стоимость маршрута (бензин + аренда ТС) 50 524 руб.

На рис. 7 представлен результирующий маршрут двух ТС типа 3 из депо Москва.

Общая стоимость маршрута: 129 650 руб.

3. Депо Владивосток.

Маршрут 5 (используется ТС типа 2):

Владивосток—Находка—Хабаровск—Комсомольск-на-Амуре—Владивосток.

Длина маршрута 2620 км.

Период планирования 3 дня.

Загруженность ТС 8328,13 кг.

Стоимость маршрута (бензин + аренда ТС) 64 103 руб.

Маршрут 6 (используется ТС типа 2):

Владивосток—Иркутск—Улан-Удэ—Владивосток.

Длина маршрута 7801 км.

Период планирования 7 дней.

Загруженность ТС 3402,815 кг.

Стоимость маршрута (бензин + аренда ТС) 190 865 руб.

Общая стоимость маршрута 254 968 руб.

В рассмотренном примере рассматривались следующие ограничения:

- множество депо (Уфа, Москва, Владивосток);
- раздельная доставка (Набережные Челны);
- временные окна (Самара, Саратов, Нижний Новгород, Казань, Воронеж, Волгоград, Сызрань);
- возврат товаров (Находка, Хабаровск, Комсомольск-на-Амуре, Улан-Удэ, Иркутск);
- период планирования (7 дней для всех клиентов депо Владивосток);
- учет рационального размещения груза внутри ТС при построении рациональных маршрутов доставки.

Таким образом, результаты, полученные алгоритмом P-ACO-EVRP, позволяют повысить эффективность доставки однородного груза различным клиентам, снизив затраты на доставку груза в среднем на 12 %.

Заключение

В работе приведены результаты численных экспериментов алгоритма P-ACO-EVRP для решения задачи оптимизации доставки однородного груза (заказа) различным клиентам по критерию минимума стоимости пройденного транспортными средствами пути. В силу новизны рассматриваемой задачи доставки однородного груза (заказа) различным клиентам не существует как наборов тестовых данных, так и результатов других методов решения для проведения сравнительного анализа. При этом были приведены расчеты маршрутов на реальных примерах, и их сравнивали с результатами известного ПО logvnp, которые показали преимущество предложенного алгоритма. При поиске рациональных маршрутов учитывали такие ограничения, как грузоподъемность транспортного средства, временные окна, период планирования, множество депо, раздельная доставка, неоднородный парк ТС, возможность возврата груза, рацио-

нальное размещение груза (заказа) в ТС во время построения рациональных маршрутов доставки.

Список литературы

1. **Angus D.** Population-Based Ant Colony Optimisation for Multi-objective Function Optimisation // *Proceedings ACAL*, 2007. P. 232–244.
2. **Dorigo M., Gambardella L. M.** Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem // *IEEE Transactions on Evolutionary Computation*. 1997. V. 1. P. 53–66.
3. **Gendreau M.** Metaheuristics for the vehicle routing problem. Technical Report CRT-963, Centre de Recherche sur les Transports, Universit de Montral, 1994.
4. **Ho S. C., Haugland D.** A tabu search heuristic for the vehicle routing problem with time windows and split deliveries // *Computers & Operations Research*. 2004. Vol. 31, N. 12. P. 1947–1964.
5. **Lenstra J., Rinnooy K.** Complexity of vehicle routing and scheduling problems // *Networks*. 1981. Vol. 11. N. 2. P. 221–227.
6. **Li H., Lim A.** A Metaheuristic for Solving the Pickup and Delivery Problem with Time Windows // *IEEE Computer Society (ed.) Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE Computer Society, Los Alamitos, California, 2001. P. 160–167.
7. **Luke S.** Essentials of Metaheuristics. Raleigh: Lulu, 2009. 235 p.
8. **Pirkwieser S., Raidl G. R.** Multiple Variable Neighborhood Search Enriched with ILP Techniques for the Periodic Vehicle Routing Problem with Time Windows // *Hybrid Metaheuristics*. Berlin, Heidelberg: Springer-Verlag, 2009. Vol. 5818. P. 45–59.
9. **Solomon M. M.** Algorithms for the vehicle routing and scheduling problem with time window constraints // *European Journal Operation Research*. 1987. Vol. 35, N. 2. P. 254–265.
10. **Toth P., Vigo D.** An overview of vehicle routing problems. Philadelphia: SIAM, 2001. P. 1–26.
11. <https://logvnp.com/logvrpsite/en/index.html>

A. F. Valeeva, D. Sc., Professor, aida_val2004@mail.ru, Ju. A. Goncharova, Postgraduate, Ufa State Aviation Technical University

The Development of the Logistic Transport System

The operational planning problems solving of homogeneous goods delivery to different customers such as inventory management and looking for rational routes of goods delivery are considered in this article. The algorithm for searching the best inventory management strategy and a modified population based ant colony optimization algorithm (P-ACO) for getting the rational routes of homogeneous goods delivery to different customers are presented. Some numerical experiments on random generated data, taking into account such restrictions as vehicle capacity, time windows, split delivery, delivery period, multi depot, pickup and delivery, and tested examples, taken from International OR-library tests, are resulted. Numerical experiments results confirm the effectiveness of the developed algorithms based on the P-ACO algorithm. The problem of rational goods packing in vehicles (the problem of three-dimensional packing) solved in the constructing routes process.

Keywords: inventory management, routing, goods packing in vehicles, population based ant colony optimization algorithm

References

1. **Angus D.** Population-Based Ant Colony Optimisation for Multi-objective Function Optimisation, *Proceedings ACAL*, 2007, pp. 232–244.
2. **Dorigo M., Gambardella L. M.** Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, 1997, vol. 1, pp. 53–66.
3. **Gendreau M.** Metaheuristics for the vehicle routing problem, Technical Report CRT-963, Centre de Recherche sur les Transports, Universit de Montral, 1994.
4. **Ho S. C., Haugland D.** A tabu search heuristic for the vehicle routing problem with time windows and split deliveries, *Computers & Operations Research*, 2004, vol. 31, no. 12, pp. 1947–1964.
5. **Lenstra J., Rinnooy K.** Complexity of vehicle routing and scheduling problems. *Networks*, 1981, vol. 11, no. 2, pp. 221–227.
6. **Li H., Lim A.** A Metaheuristic for Solving the Pickup and Delivery Problem with Time Windows, *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE Computer Society, Los Alamitos, California, 2001, pp. 160–167.
7. **Luke S.** Essentials of Metaheuristics, Raleigh, Lulu, 2009, 235 p.
8. **Pirkwieser S., Raidl G. R.** Multiple Variable Neighborhood Search Enriched with ILP Techniques for the Periodic Vehicle Routing Problem with Time Windows. *Hybrid Metaheuristics*. Berlin Heidelberg, Springer-Verlag, 2009, vol. 5818, pp. 45–59.
9. **Solomon M. M.** Algorithms for the vehicle routing and scheduling problem with time window constraints, *European Journal Operation Research*, 1987, vol. 35, no. 2, pp. 254–265.
10. **Toth P., Vigo D.** An overview of vehicle routing problems, *Philadelphia; SIAM*, 2001, 367 p.
11. <https://logvnp.com/logvrpsite/en/index.html>

Н. П. Деменков, канд. техн. наук, доц., e-mail: dnp@bmstu.ru,
Е. А. Микрин, д-р техн. наук, проф., e-mail: evgeny.mikrin@bmstu.ru,
И. А. Мочалов, д-р техн. наук, проф., e-mail: intelsyst@mail.ru,
 МГТУ им. Н. Э. Баумана, Москва

Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть 1

Определяются обобщенные нечеткие производные Хукухары (Hukuhara) различных типов и на их основе даются определения прямого и обратного нечеткого преобразования Лапласа. Указываются отличия его от традиционного преобразования. Определяются типы и их число нечетких дифференциальных уравнений, которые решаются путем применения нечеткого преобразования Лапласа.

Ключевые слова: нечеткая краевая задача, нечеткое преобразование Лапласа, нечеткое прямое и обратное преобразование Лапласа, нечеткая производная по Хукухара

Введение

В практической и научной деятельности широко используется математическое моделирование, состоящее в представлении причинно-следственных связей реальных процессов в виде математических моделей и их решений. При разработке математических моделей к ним обычно предъявляются требования адекватности и простоты реализации. Адекватность подразумевает учет в модели различного рода неопределенностей, а простота определяется методами нахождения решений по этим моделям. Этим требованиям в полной мере отвечают нечеткие математические модели. Их условно можно разделить на два класса.

Первый класс объединяет совокупность нечетких логических моделей. Они имеют структуру "если—то". Вместе с заданием функций принадлежности (операция фаззификации — fz), логической обработкой фаззифицированных переменных (нечеткие логические операции: "И", "ИЛИ", "НЕ" и др.) и процедурой дефаззификации (dfz) они реализуют нечеткий логический контроллер (вычислитель). Его функционирование обеспечивается нечеткой базой правил, входящей в состав системного программного обеспечения контроллера, с помощью которого обрабатывается информация от объекта управления и формируются управляющие команды для него. В настоящее время техника нечеткого логического управления достаточно хорошо отработана и широко применяется в системах логического управления различных типов [1, 2]. Нечеткая логика нашла свое применение в традиционных системах управления путем интеллектуализации классических ПИД и нелинейных регуляторов, создания адаптивных и других типов регуляторов [3, 4].

Второй класс моделей представляется традиционными математическими моделями в нечеткой их реализации, что обеспечивает им адекватность при

воздействии различного рода возмущений. В традиционной математике при нахождении решений с помощью классических моделей в их четкой реализации широко используется операционный метод, в котором исходная модель преобразуется в операторное уравнение [5, 6]. При его решении используются два подхода. В первом задача нахождения решения формируется в виде поиска некоторого базиса, в котором оператор уравнения имеет наиболее простое представление. Это приводит к спектральной теории операторов. Второй подход состоит в нахождении нового пространства, изоморфного исходному, в котором оператор описывается наиболее простым способом. Часто в качестве изоморфного преобразования используется интегральное преобразование Лапласа. Обычно использование интегралов специальных видов приводит к интегральным преобразованиям различных типов, например, интегральные преобразования Меллина, Фурье, Вебера, Ганкеля и др. Современное операционное исчисление применительно к линейным дифференциальным уравнениям имеет типичную схему интегральных преобразований, т. е. имеется прямое интегральное преобразование, когда пространство оригиналов (дифференциальные уравнения) трансформируется в пространство изображений (алгебраические уравнения). После решения последнего с использованием обратного преобразования находится решение исходного уравнения. Эта схема является типичной, например, в элементарной математике преобразование $\ln - \ln^{-1}$, или в нечетком логическом управлении " $fz - dfz$ ".

В современной теории управления важное место занимают нечеткие модели, которые описывают линейные и нелинейные объекты управления, датчики, исполнительные механизмы и другие компоненты систем управления. Их описание в условиях неопределенности приводит к различным типам нечетких моделей: нечеткая начальная задача для линейных

и нелинейных дифференциальных уравнений с постоянными и переменными коэффициентами; нечеткие одно- и двухточечные граничные задачи для дифференциальных уравнений первого и второго порядков; нечеткие интегральные уравнения Вольтерра типа свертки 1-го и 2-го рода; нечеткие уравнения в частных производных второго порядка; нечеткие интегро-дифференциальные уравнения и др.

Решение этих уравнений тесно связано с представлением производной в нечеткой интерпретации. Одними из первых авторов в представлении нечеткой начальной задачи являются работы О. Kaleva [7] и S. Seikkala [8]. На основе параметрической формы представления нечетких переменных, нечеткое уравнение первого порядка в этих работах представляется в виде традиционной системы дифференциальных уравнений второго порядка, которое решается матричным методом. Другими работами в этом направлении являются работы J. J. Buckley и M. M. Friedman [9–11]. В этих работах была показана эквивалентность многих типов нечетких производных, связанных с заданием метрик в нечетком векторном пространстве и способом задания в нем арифметических операций. Кроме того, была показана неединственность решения нечеткой начальной задачи и найдена связь между этими решениями. При дальнейшей реализации этого направления, в частности, решения нечеткой начальной задачи второго, третьего и других порядков возникли трудности в представлении нечетких производных выше первого порядка. Выход из этого затруднения был найден путем перехода к обобщенной нечеткой производной H (Hukuhara)-типа [12]. В рамках этого перехода были реализованы нечеткие левые и правые производные и далее были определены нечеткие производные H -типа второго, третьего и других порядков. Это позволило сформулировать нечеткую начальную задачу n -го порядка. Для ее решения в линейном случае, как это обычно принято в инженерной практике, было предложено использование интегрального преобразования в виде нечеткого преобразования Лапласа [13–16]. Оно было предложено в работе [16] и нашло широкое применение для решения перечисленных выше нечетких линейных дифференциальных уравнений различных типов.

Нечеткое преобразование Лапласа (L) имеет принципиальное отличие от его традиционного аналога. Так, изображение по L имеет аргумент в виде действительной переменной и соответствующее нечеткое изображение $F(s)$ также является действительной переменной, тогда как в традиционном преобразовании изображение $F(s)$ определяется функцией комплексного переменного, которая аналитична в полуплоскости $\text{Res} > \sigma_0$, где σ_0 — показатель роста оригинала. Другим отличием нечеткого L преобразования является отсутствие для него обратного преобразования, заданного в виде неко-

торой формулы, как это принято в традиционных преобразованиях. Формальное совпадение формул прямого преобразования Лапласа для четкого (традиционного) и нечеткого случаев приводит к совпадению основных свойств этого преобразования. Это позволяет использовать традиционные таблицы простейших оригиналов и их изображений для решения разнообразных нечетких уравнений. В случае отсутствия в этих таблицах стандартных оригиналов и изображений применяется метод дифференциального преобразования (D) [17]. Это позволяет реализовывать нестандартные ситуации с таблицами. Метод D , в частности, позволяет по нечеткому изображению находить нечеткий оригинал.

Помимо L - и D -преобразований также широко используется нечеткое преобразование F (fuzzy) [18].

1. Постановка задачи

Цель настоящей работы состоит в обобщении представлений нечеткого преобразования Лапласа, его свойств и применение его для решения перечисленных выше разнообразных нечетких линейных и нелинейных задач. Для линеаризации последних будет использован метод декомпозиции, который будет изложен в части II.

2. Нечеткое преобразование Лапласа

Полагаем, что имеют место следующие базовые определения, приведенные в работах [9–11], которые будут использованы далее: функция принадлежности $r: x \in R_1 \rightarrow r \in [0; 1] \subset R_1$; нечеткое число и его типы; нечеткое отображение (функция) $f_H(x)$, где "н" — индекс нечеткости; параметрическая форма представления нечеткой функции $f_H(x) = f(x, r) = (\underline{f}(x, r), \overline{f}(x, r))$, $r \in [0; 1]$, где $\underline{f}(x, r)$, $\overline{f}(x, r)$ — миноранта и мажоранта нечеткой функции f_H ; нечеткое векторное пространство; метрика Хаусдорфа; типы нечетких производных и их взаимосвязь и т. д. Дальнейшие определения и теоремы будут вводиться по мере их необходимости.

Будем полагать, что $f_H(x)$ является непрерывной нечеткой функцией и $s > 0$ — действительный параметр $s \in R_1$; полагается также, что $f_H(x)\exp(-sx)$ имеет нечеткую несобственную интегрируемость по Риману на $[0, \infty) \subset R_1$, тогда

$$\int_0^{\infty} f_H(x) e^{-sx} dx = \lim_{x \rightarrow \infty} \int_0^x f_H(x) e^{-sx} dx$$

называется нечетким преобразованием Лапласа L для нечеткой функции $f_H(x)$ и обозначается $L[f_H(x)] = F_H(s)$.

Здесь $f_H(x)$ называется нечетким оригиналом, а $F_H(s)$ — нечетким изображением по Лапласу. Полагается, что $f_H(x)$ имеет ограниченный порядок роста. В параметрической форме задания $f_H(x)$ имеем $L[f_H(x)] = L[f(x, r)] = [l(\underline{f}(x, r)), l(\overline{f}(x, r))]$, $r \in [0; 1]$,

где

$$l(\underline{f}(x, r)) = \int_0^{\infty} \underline{f}(x, r) e^{-sx} dx = \lim_{x \rightarrow \infty} \int_0^x \underline{f}(x, r) e^{-sx} dx;$$

$$l(\overline{f}(x, r)) = \int_0^{\infty} \overline{f}(x, r) e^{-sx} dx = \lim_{x \rightarrow \infty} \int_0^x \overline{f}(x, r) e^{-sx} dx.$$

Обратное нечеткое преобразование L^{-1} для нечеткого изображения $F_H(s)$ определяется в виде

$$L^{-1}[F_H(s)] = f_H(x),$$

и следует отметить, что оно не имеет в явном виде формульного выражения, как это принято в традиционном преобразовании Лапласа.

По аналогии с традиционным преобразованием [19] для нечеткого преобразования L имеют место следующие очевидные свойства, которые перечислим без доказательства:

- линейность;
- смещение оригинала и изображения;
- изменение масштабов оригинала и изображения;
- свертка оригиналов (умножений изображений);
- дифференцирование и интегрирование изображений.

Важным свойством преобразования L является нечеткое дифференцирование оригинала. В традиционном преобразовании это свойство можно записать в виде

$$L[f^{(n)}(x)] = s^n F(s) - \sum_{k=1}^n s^{n-k} f^{(k-1)}(x=+0), \quad (1)$$

из которой следует необходимость в определении правой четкой производной $f^{(k-1)}(x=+0)$ порядка $(k-1)$ в окрестности нуля и производной порядка выше первого $f^{(n)}(x)$.

Следовательно, в общем случае при переходе к нечеткому аналогу этой формулы возникает необходимость в определении нечетких производных порядка выше первого. Для них производные соответствующих порядков типа Сейккалы, Дубоиса и т. д., представленные в работах [9–11], не определены, поэтому свойство (1) в нечетком случае отсутствует. Выходом из этой ситуации является использование H -разности (Hukuhara — H -difference), обобщенной H -дифференцируемости в точке $x = x_0$ по Хукухары и сильной (strongly) дифференцируемости gH по Хукухары.

Определение 1. Пусть $u, v \in E$, где E — нечеткое множество. Если существует $w \in E$: $u = v + w$, то w называется H -разностью и обозначается $u \ominus v = w$. Эта разность предполагает, что $u \ominus v \neq u + (-v)$, т. е. v не является обратным элементом.

Определение 2 [10]. Пусть имеем нечеткое отображение $f: (a, b) \rightarrow R_1$ и $x_0 \in (a, b) \subset R_1$. Тогда f является H -дифференцируемой в точке x_0 , если су-

ществует элемент $\dot{f}(x_0) \in R_1$ такой, что для малого $h > 0 \in R_1$ существуют:

$$f(x_0 + h) \ominus f(x_0), f(x_0) \ominus f(x_0 - h)$$

и их пределы в метрике L Хаусдорфа определяются выражениями

$$\begin{aligned} & \lim_{h \rightarrow 0^+} [(f(x_0 + h) \ominus f(x_0))/h] = \\ & = \lim_{h \rightarrow 0^-} [(f(x_0) \ominus f(x_0 - h))/h] = \dot{f}(x_0). \end{aligned}$$

Определение 3 [10]. Пусть имеем отображение $f: (a, b) \rightarrow R_1$ и $x_0 \in (a, b) \subset R_1$, тогда f называется сильно обобщенной gH -дифференцируемой в точке x_0 , если существует элемент $\dot{f}(x_0) \in R_1$ такой, что

$$(i) \text{ для малого } h > 0 \in R_1 \text{ существуют} \\ f(x_0 + h) \ominus f(x_0), f(x_0) \ominus f(x_0 - h)$$

и их пределы в метрике L определяются формулами

$$\begin{aligned} & \lim_{h \rightarrow 0^+} [(f(x_0 + h) \ominus f(x_0))/h] = \\ & = \lim_{h \rightarrow 0^-} [(f(x_0) \ominus (f(x_0 - h)))/h] = \dot{f}(x_0); \end{aligned}$$

(ii) или для малого $h > 0 \in R_1$ существуют

$$f(x_0 + h) \ominus f(x_0), f(x_0 - h) \ominus f(x_0)$$

и их пределы в метрике L равны

$$\begin{aligned} & \lim_{h \rightarrow 0^+} [(f(x_0) \ominus f(x_0 + h))/(-h)] = \\ & = \lim_{h \rightarrow 0^-} [(f(x_0 - h) \ominus (f(x_0)))/(-h)] = \dot{f}(x_0), \end{aligned}$$

(iii) или для малого $h > 0 \in R_1$ существуют

$$f(x_0 + h) \ominus f(x_0), f(x_0 - h) \ominus f(x_0)$$

и их пределы в метрике L равны

$$\begin{aligned} & \lim_{h \rightarrow 0^+} [(f(x_0 + h) \ominus f(x_0))/h] = \\ & = \lim_{h \rightarrow 0^-} [(f(x_0 - h) \ominus (f(x_0)))/(-h)] = \dot{f}(x_0), \end{aligned}$$

(iv) или для малого $h > 0 \in R_1$ существуют

$$f(x_0 + h) \ominus f(x_0), f(x_0 - h) \ominus f(x_0)$$

и их пределы в метрике L равны

$$\begin{aligned} & \lim_{h \rightarrow 0^+} [(f(x_0 + h) \ominus f(x_0))/(-h)] = \\ & = \lim_{h \rightarrow 0^-} [(f(x_0 + h) \ominus (f(x_0)))/h] = \dot{f}(x_0). \end{aligned}$$

Если существует gH -дифференцируемость $f(x)$ для $\forall x \in (a, b)$, то $f(x)$ является gH -дифференцируемой на интервале (a, b) .

Пусть функция $f_H(x)$ представлена в параметрической форме

$$f_H(x) = f(x, r) = (\underline{f}(x, r), \overline{f}(x, r)), \quad r \in [0; 1].$$

Тогда из определений (i), (ii) gH -дифференцируемости следует:

$\dot{f}(x, r) = (\dot{\underline{f}}(x, r), \dot{\overline{f}}(x, r)) - gH_{(i)}$ производная первого порядка;

$\dot{f}(x, r) = (\dot{\bar{f}}(x, r), \dot{\underline{f}}(x, r)) - gH_{(ii)}$ производная первого порядка.

На основании теоремы о дифференцировании нечеткого оригинала $f_H(x)$ в этом случае имеем:

$L[\dot{f}(x, r)] = sL[f(x, r)] \ominus f(0)$ — для $gH_{(i)}$ производной;

$L[\dot{\bar{f}}(x, r)] = -sL[\dot{f}(x, r)] \ominus (-f(0))$ — для $gH_{(ii)}$ производной.

По аналогии с предыдущим можно получить вторую нечеткую производную $\ddot{f}_H(x)$. Для этого в (i)–(iv) необходимо заменить \dot{f} на \ddot{f} . В результате получим:

$\ddot{f}(x, r) = (\ddot{\bar{f}}(x, r), \ddot{\underline{f}}(x, r)) - gH_{(i)}$ производная второго порядка;

$\ddot{\bar{f}}(x, r) = (\ddot{\bar{f}}(x, r), \ddot{\underline{f}}(x, r)) - gH_{(ii)}$ производная второго порядка.

Нечеткие преобразования соответственно будут равны следующим случаям:

а) $L[\ddot{f}(x, r)] = s^2L[f(x, r)] \ominus sf(0) \ominus \dot{f}(0)$ для $f(x, r)$, $\dot{f}(x, r)$ $gH_{(i)}$ дифференцируемых;

б) $L[\ddot{\bar{f}}(x, r)] = -s^2L[\ddot{f}(x, r)] \ominus (-sf(0)) \ominus (-\dot{f}(0))$ для $f(x, r)$ $gH_{(i)}$ дифференцируемой, для $\dot{f}(x, r)$ $gH_{(ii)}$ дифференцируемой;

в) $L[\ddot{f}(x, r)] = -s^2L[\ddot{f}(x, r)] \ominus (-sf(0)) \ominus (\dot{f}(0))$ для $f(x, r)$ $gH_{(ii)}$ дифференцируемой, для $\dot{f}(x, r)$ $gH_{(i)}$ дифференцируемой;

г) $L[\ddot{\bar{f}}(x, r)] = s^2L[\ddot{f}(x, r)] \ominus sf(0) \ominus (-f(0))$ для $f(x, r)$, $\dot{f}(x, r)$ $gH_{(ii)}$ дифференцируемых.

Получая по аналогичной методике $\ddot{\bar{f}}(x, r)$, ..., $f^{(k)}(x, r)$ можно найти нечеткое преобразование Лапласа производных более высоких порядков $L[\ddot{\bar{f}}(x, r)]$, ..., $L[f^{(k)}(x, r)]$.

Общее число m случаев преобразований с учетом возможных комбинаций $gH_{(i)}$, $gH_{(ii)}$ равно $m = 2^n$, где n — порядок производной. Например, при $n = 2$ $m = 2^2 = 4$, т.е. случаи а)–д); при $n = 3$ $m = 2^3 = 8$ случаев и т.д.

3. Решение нечеткой начальной задачи методом нечеткого преобразования Лапласа

Рассмотрим решения нечетких начальных задач первого и второго порядков.

3.1. Нечеткая начальная задача первого порядка [11, 20]

Имеем нечеткую начальную задачу первого порядка:

$$\dot{y}(x, r) + p(r)y(x, r) = a(r),$$

в которой нечеткие треугольные числа a и p зависят от параметра r и определяются посредством трех чисел соответственно:

$$a(r) = (a_1/a_2/a_3), a_1 < a_2 < a_3, a_i > 0,$$

$$p(r) = (p_1/p_2/p_3), p_1 < p_2 < p_3, p_i > 0,$$

с начальными условиями для y при $x = 0$:

$$y(x = 0, r) = \underline{y}(0, r) = b_1 + (b_2 - b_1)r,$$

$$\bar{y}(0, r) = b_3 - (b_2 - b_3)r, r \in [0; 1].$$

Необходимо найти решение этой задачи.

Если $y(x)$ является H -дифференцируемой, то имеем

$$\dot{\underline{y}}(x, r) + \underline{p}(r)\underline{y}(x, r) = \underline{a}(r),$$

$$\underline{a}(r) = a_1 + (a_2 - a_1)r, \underline{p}(r) = p_1 + (p_2 - p_1)r,$$

$$\underline{y}(0, r) = b_1 + (b_2 - b_1)r,$$

$$\dot{\bar{y}}(x, r) + \bar{p}(r)\bar{y}(x, r) = \bar{a}(r), \bar{a}(r) = a_3 + (a_2 - a_1)r,$$

$$\bar{p}(r) = p_3 + (p_2 - p_1)r,$$

$$\bar{y}(0, r) = b_3 + (b_2 - b_3)r.$$

Поэтому преобразование L дает:

$$sL[\underline{y}(x, r)] \ominus \underline{y}(0, r) + \underline{p}(r)L[\underline{y}(x, r)] = L[\underline{a}(r)];$$

$$sL[\bar{y}(x, r)] \ominus \bar{y}(0, r) + \bar{p}(r)L[\bar{y}(x, r)] = L[\bar{a}(r)],$$

$$(s + \underline{p}(r))L[\underline{y}(x, r)] = \underline{y}(0, r) + \underline{a}(r)/s;$$

$$(s + \bar{p}(r))L[\bar{y}(x, r)] = \bar{y}(0, r) + \bar{a}(r)/s,$$

$$L[\underline{y}(x, r)] = \underline{y}(0, r)(s + \underline{p}(r))^{-1} + \underline{a}(r)(s + \underline{p}(r))^{-1};$$

$$L[\bar{y}(x, r)] = \bar{y}(0, r)(s + \bar{p}(r))^{-1} + \bar{a}(r)(s + \bar{p}(r))^{-1},$$

откуда после применения L^{-1} и табличных значений получим нечеткое решение исходной нечеткой задачи:

$$y_H(x) = y(x, r) = (\underline{y}(x, r), \bar{y}(x, r)) = \underline{y}(0, r) - \underline{a}(r)e^{-\underline{p}(r)x} +$$

$$+ \underline{a}(r), \bar{y}(x, r) = \bar{y}(0, r) - \bar{a}(r)e^{-\bar{p}(r)x} + \bar{a}(r).$$

При $r = 1$ имеем

$$\underline{a}(r = 1) = \bar{a}(r = 1) = a_2; \underline{p}(r = 1) = \bar{p}(r = 1) = p_2;$$

$$\underline{y}(0, r) = \bar{y}(0, r) = b_2.$$

Следовательно,

$$\dot{y}(x) + p_2y(x) = a_2, y(x = 0) = b_2$$

$$\text{и } y(x) = (b_2 - a_2)e^{-p_2x}.$$

Последнее уравнение является частным случаем его нечеткого аналога.

3.2. Нечеткая начальная задача второго порядка [11, 21]

Имеем нечеткую начальную задачу второго порядка

$$\ddot{y}(x) = f(x, y(x), \dot{y}(x))$$

с начальными условиями для y и \dot{y} при $x = 0$:

$$y(x=0) = y_0(r) = (\underline{y}_0(r), \bar{y}_0(r)),$$

$$\dot{y}(x=0) = z_0(r) = (\underline{z}_0(r), \bar{z}_0(r)).$$

Здесь $y_0(r), z_0(r)$ — нечеткие переменные, $r \in [0, 1]$, а $f(\cdot)$ — нечеткая линейная функция относительно $y(x), \dot{y}(x)$ и x . Необходимо найти решение.

Используя преобразование L , получим:

$$L[\ddot{y}(x)] = L[f(x, y(x), \dot{y}(x))].$$

В результате будем иметь следующие альтернативные уравнения.

Случай 1. Если $y(x), \dot{y}(x)$ $gH_{(i)}$ -дифференцируемы, т. е.

$$\dot{y}(x, r) = (\underline{\dot{y}}(x, r), \bar{\dot{y}}(x, r)),$$

$$\ddot{y}(x, r) = (\underline{\ddot{y}}(x, r), \bar{\ddot{y}}(x, r)), \text{ то}$$

$$L[\ddot{y}(x)] = L[f(x, y(x), \dot{y}(x))] \Rightarrow$$

$$\Rightarrow s^2 L[y(x)] \ominus s y_0(r) \ominus z_0(r) = L[f(x, y, \dot{y})] \Rightarrow$$

$$\Rightarrow \begin{cases} s^2 L[\underline{y}(x, r)] - s \underline{y}_0(r) - \underline{z}_0(r) = L[\underline{f}(\cdot)]; \\ s^2 L[\bar{y}(x, r)] - s \bar{y}_0(r) - \bar{z}_0(r) = L[\bar{f}(\cdot)]; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} L[\underline{y}(x, r)] = H_1(s, r); \\ L[\bar{y}(x, r)] = K_1(s, r); \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = L^{-1}[H_1(s, r)]; \\ \bar{y}(x, r) = L^{-1}[K_1(s, r)]. \end{cases}$$

Случай 2. Если $y(x)$ — $gH_{(i)}$ -дифференцируема, а $\dot{y}(x)$ — $gH_{(ii)}$ -дифференцируема, т. е.

$$\dot{y}(x, r) = (\underline{\dot{y}}(x, r), \bar{\dot{y}}(x, r)),$$

$$\ddot{y}(x, r) = (\underline{\ddot{y}}(x, r), \bar{\ddot{y}}(x, r)), \text{ то}$$

$$L[\ddot{y}(x)] = -\dot{y}(0) \ominus \{-s^2 L[y(x, r)] \ominus (-s y_0(0))\} \Rightarrow$$

$$\Rightarrow (-z_0(r)) \ominus \{-s^2 L[y(x, r)] \ominus (-s y_0(0))\} = L[f(\cdot)] \Rightarrow$$

$$\Rightarrow \begin{cases} s^2 L[\underline{y}(x, r)] - s \underline{y}_0 - \underline{z}_0(r) = L[\underline{f}(\cdot)]; \\ s^2 L[\bar{y}(x, r)] - s \bar{y}_0 - \bar{z}_0(r) = L[\bar{f}(\cdot)]; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} L[\underline{y}(x, r)] = H_2(s, r); \\ L[\bar{y}(x, r)] = K_2(s, r); \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = L^{-1}[H_2(s, r)]; \\ \bar{y}(x, r) = L^{-1}[K_2(s, r)]. \end{cases}$$

Случай 3. Если $y(x)$ — $gH_{(ii)}$ -дифференцируема, а $\dot{y}(x)$ — $gH_{(i)}$ -дифференцируема, т. е.

$$\dot{y}(x, r) = (\underline{\dot{y}}(x, r), \bar{\dot{y}}(x, r)),$$

$$\ddot{y}(x, r) = (\underline{\ddot{y}}(x, r), \bar{\ddot{y}}(x, r)), \text{ то}$$

$$L[\ddot{y}(x)] = -s y_0(0) \ominus \{-s^2 L[y(x)]\} \ominus \dot{y}(0) \Rightarrow$$

$$\Rightarrow \begin{cases} s^2 L[\underline{y}(x, r)] - s \underline{y}_0(r) - \underline{z}_0(r) = L[\underline{f}(\cdot)]; \\ s^2 L[\bar{y}(x, r)] - s \bar{y}_0(r) - \bar{z}_0(r) = L[\bar{f}(\cdot)]; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} L[\underline{y}(x, r)] = H_3(s, r); \\ L[\bar{y}(x, r)] = K_3(s, r); \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = L^{-1}[H_3(s, r)]; \\ \bar{y}(x, r) = L^{-1}[K_3(s, r)]. \end{cases}$$

Случай 4. Если $y(x), \dot{y}(x)$ — $gH_{(ii)}$ -дифференцируемы, т. е.

$$\dot{y}(x, r) = (\underline{\dot{y}}(x, r), \bar{\dot{y}}(x, r)),$$

$$\ddot{y}(x, r) = (\underline{\ddot{y}}(x, r), \bar{\ddot{y}}(x, r)), \text{ то}$$

$$\begin{cases} s^2 L[\underline{y}(x, r)] - s \underline{y}_0(r) - \underline{z}_0(r) = L[\underline{f}(\cdot)]; \\ s^2 L[\bar{y}(x, r)] - s \bar{y}_0(r) - \bar{z}_0(r) = L[\bar{f}(\cdot)]; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} L[\underline{y}(x, r)] = H_4(s, r); \\ L[\bar{y}(x, r)] = K_4(s, r); \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = L^{-1}[H_4(s, r)]; \\ \bar{y}(x, r) = L^{-1}[K_4(s, r)]. \end{cases}$$

Величины $H_i(\cdot), K_i(\cdot), i = 1, \dots, 4$, могут быть легко вычислены при линейности $f(\cdot)$ относительно $y(\cdot), \dot{y}(\cdot)$ и представлены в виде отношений многочленов относительно s [13].

Пример. Пусть имеем нечеткую начальную задачу второго порядка в параметрической форме:

$$\ddot{y}(x) + y(x) = a_H, \quad a_H = a(r) = (\underline{a}(r), \bar{a}(r)),$$

$$\bar{a}(r) = 2 - r, \quad r \in [0, 1];$$

$$y(x=0, r) = y_0(r) = (\underline{y}_0(r) = r - 1, \bar{y}_0(r) = 1 - r);$$

$$\dot{y}(x=0, r) = \dot{y}_0(r) = (\underline{\dot{y}}_0(r) = r - 1, \bar{\dot{y}}_0(r) = 1 - r).$$

Рассмотрим четыре варианта решения.

Для *случая 1*, когда $y(x), \dot{y}(x)$ — обе $gH_{(i)}$ -дифференцируемы, поэтому:

$$\begin{aligned} & \begin{cases} \underline{\ddot{y}}(x, r) + \underline{y}(x, r) = r; \\ \bar{\ddot{y}}(x, r) + \bar{y}(x, r) = 2 - r; \end{cases} \Rightarrow \\ & \Rightarrow \begin{cases} L[\underline{\ddot{y}}(x, r)] + L[\underline{y}(x, r)] = \frac{r}{s}; \\ L[\bar{\ddot{y}}(x, r)] + L[\bar{y}(x, r)] = \frac{2-r}{s}; \end{cases} \Rightarrow \\ & \Rightarrow \begin{cases} L[\underline{y}(\cdot)] = (r-1) \frac{s+1}{s^2+1} + r \left(\frac{1}{s} - \frac{s}{s^2+1} \right); \\ L[\bar{y}(\cdot)] = (1-r) \frac{s+1}{s^2+1} + (2-r) \left(\frac{1}{s} - \frac{s}{s^2+1} \right); \end{cases} \Rightarrow \\ & \Rightarrow \begin{cases} \underline{y}(x, r) = r(1 + \sin x) - \sin x - \cos x; \\ \bar{y}(x, r) = (2-r)(1 + \sin x) - \sin x - \cos x. \end{cases} \end{aligned}$$

Таблица 1

Операторная форма $gH_{(i)}$, $gH_{(ii)}$ -дифференцируемости второго порядка

Типы нечетких уравнений	Операторы D_1, D_2 второго порядка
Случай 1	$(D_1, D_1)f$
Случай 2	$(D_1, D_2)f$
Случай 3	$(D_2, D_1)f$
Случай 4	$(D_2, D_2)f$

Таблица 2

Операторная форма $gH_{(i)}$, $gH_{(ii)}$ -дифференцируемости четвертого порядка

Тип	Оператор	Тип	Оператор	Тип	Оператор	Тип	Оператор
1	$(D_1, D_1, D_1, D_1)f$	5	$(D_2, D_2, D_2, D_2)f$	9	$(D_1, D_2, D_1, D_2)f$	13	$(D_1, D_1, D_2, D_1)f$
2	$(D_2, D_1, D_1, D_1)f$	6	$(D_1, D_2, D_2, D_2)f$	10	$(D_2, D_1, D_2, D_1)f$	14	$(D_2, D_1, D_2, D_2)f$
3	$(D_2, D_2, D_1, D_1)f$	7	$(D_1, D_1, D_2, D_2)f$	11	$(D_2, D_2, D_1, D_2)f$	15	$(D_2, D_1, D_1, D_2)f$
4	$(D_2, D_2, D_2, D_1)f$	8	$(D_1, D_1, D_1, D_2)f$	12	$(D_1, D_2, D_1, D_1)f$	16	$(D_1, D_2, D_2, D_1)f$

Для случая 2, когда $y(x)$ — $gH_{(i)}$ -дифференцируема, а $\dot{y}(x)$ — $gH_{(ii)}$ -дифференцируема, поэтому:

$$\begin{cases} I[\underline{\bar{y}}(x, r)] + I[\underline{y}(x, r)] = \frac{r}{s}; \\ I[\underline{\dot{y}}(x, r)] + I[\underline{\bar{y}}(x, r)] = \frac{2-r}{s}; \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = r(1 + \operatorname{sh}x) - \operatorname{sh}x - \operatorname{cos}x; \\ \underline{\bar{y}}(x, r) = (2-r)(1 + \operatorname{sh}x) - \operatorname{sh}x - \operatorname{cos}x. \end{cases}$$

Для случая 3, когда $y(x)$ — $gH_{(ii)}$ -дифференцируема, а $\dot{y}(x)$ — $gH_{(i)}$ -дифференцируема, поэтому:

$$\begin{cases} I[\underline{\bar{y}}(x, r)] + I[\underline{\bar{y}}(x, r)] = \frac{r}{s}; \\ I[\underline{\dot{y}}(x, r)] + I[\underline{y}(x, r)] = \frac{2-r}{s}; \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = r(1 - \operatorname{sh}x) + \operatorname{sh}x - \operatorname{cos}x; \\ \underline{\bar{y}}(x, r) = (2-r)(1 - \operatorname{sh}x) + \operatorname{sh}x - \operatorname{cos}x. \end{cases}$$

Для случая 4, когда $y(x)$, $\dot{y}(x)$ — обе $gH_{(ii)}$ -дифференцируемы, поэтому:

$$\begin{cases} I[\underline{\bar{y}}(x, r)] + I[\underline{\bar{y}}(x, r)] = \frac{r}{s}; \\ I[\underline{\dot{y}}(x, r)] + I[\underline{y}(x, r)] = \frac{2-r}{s}; \end{cases} \Rightarrow \begin{cases} \underline{y}(x, r) = r(1 - \operatorname{sin}x) + \operatorname{sin}x - \operatorname{cos}x; \\ \underline{\bar{y}}(x, r) = (2-r)(1 - \operatorname{sin}x) + \operatorname{sin}x - \operatorname{cos}x. \end{cases}$$

В работе [13] показано, что в случаях 1 и 2 решения уравнений не существует, так как для найденного $y(x)$ и затем $\dot{y}(x)$ решение не является $gH_{(i)}$ -дифференцируемой функцией. В случае 3, когда $y(x)$ — $gH_{(ii)}$ -дифференцируема, а $\dot{y}(x)$ — $gH_{(i)}$ -дифференцируема, полученное решение имеет место при $x \in [0; \ln(1 + \sqrt{2})]$. В случае 4 найденное решение справедливо для $x \in [0; \pi/2]$.

Введем в рассмотрение операторы D_1 и D_2 , определяемые $gH_{(i)}$ и $gH_{(ii)}$ типами дифференцируемости. Тогда случаи 1—4 можно представить в табличной форме (табл. 1).

Очевидно, для нечеткого дифференциального уравнения четвертого порядка ($n = 4$) будем иметь $m = 2^4 = 16$ типов уравнений, каждое из которых определяет нижнее ($\underline{\quad}$) и верхнее ($\bar{\quad}$) решения. В табл. 2 приведены все m типов дифференциальных уравнений для дифференциального оператора (уравнения) при $n = 4$ [14].

Полученные $m = 2^n$ нечетких решений из нечетких дифференциальных уравнений n -го порядка далее проверяются на выполнение для них $gH_{(i)}$ -, $gH_{(ii)}$ -дифференцируемости. Это приводит к сокращению общего числа m нечетких решений за счет исключения из них недифференцируемых решений, подобно тому, как это обычно делается для нечеткого дифференциального уравнения второго порядка.

Заключение

Даны определения нечеткого прямого и обратного преобразований Лапласа и указаны признаки его отличия от традиционного преобразования.

Определены нечеткая обобщенная производная gH типа Хукухара, нечеткие сильные обобщенные производные $gH_{(i)}$ — $gH_{(ii)}$ типов Хукухары. Сформулирована теорема о дифференцировании нечеткой производной первого порядка и выше.

Решены нечеткие начальные задачи первого и второго порядков. Показано, что в общем случае имеется $m = 2^n$ типов нечетких дифференциальных уравнений, где n — порядок дифференциального оператора. Например, при $n = 4$ имеется $m = 2^4 = 16$ типов нечетких дифференциальных уравнений, каждый тип из которых решается с помощью нечеткого преобразования Лапласа.

Приведены примеры решения простейших уравнений.

Список литературы

1. Методы робастного, нейро-нечеткого и адаптивного управления / Под общей ред. К. А. Пупкова. М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. 743 с.
2. Деменков Н. П. Нечеткое управление в технических системах. М.: Изд-во МГТУ им. Н. Э. Баумана, 2005. 198 с.
3. Ульянов С. В., Литвинцева Л. В., Добрынин В. Н., Мишин А. А. Интеллектуальное робастное управление: технология мягких вычислений. М.: ВНИИгеосистем, 2011. 408 с.
4. Пегат А. Нечеткое моделирование и управление. М.: Бинном. Лаборатория знаний, 2009. 798 с.
5. Волков Н. К., Канатников А. А. Интегральные преобразования и операционное исчисление. М.: Изд-во МГТУ им. Н. Э. Баумана, 2015. 227 с.
6. Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. М.: Физматгиз, 2006. 544 с.
7. Kaleva O. Fuzzy differential equations // Fuzzy sets and systems, 1987, 24 (3), pp. 301—317.
8. Seikkala S. On the fuzzy initial value problem // Fuzzy sets and systems, 1987, 24 (3), pp. 319—330.

9. **Friedman M., Ming M., Kandel A.** Fuzzy linear systems // Fuzzy sets and systems. 1998. Vol. 96, N. 2. P. 201–209.

10. **Buckley J. J., Feuring J.** Fuzzy differential equations // Fuzzy sets and systems. 2000. Vol. 110, N. 1. P. 43–54.

11. **Мочалов И. А., Хрисат М. С., Шихаб Еддин М. Я.** Нечеткие уравнения в частных производных в задачах управления. Информационные технологии. 2015. Т. 21, № 8. С. 563–569.

12. **Hukuhara M.** Integration des applications mesurables dont la valeur est un compact convexe (in French), Funkcialaj Ekvacioj. 1967. N. 10. P. 205–223.

13. **Sindu Devi S. and Ganesan K.** Solution of first order fuzzy differential by fuzzy Laplace transform // Global journal of pure and applied mathematics (GJPAM). 2016. Vol. 12, N. 4. P. 381–387.

14. **Eljaoui E., Melliani S. and Chadli S.** Solving second-order fuzzy differential equations by fuzzy Laplace transform method // Advances in difference equations? 2015. N. 66. P. 1–14. DOI: 10.1186/s13662-015-0414-x.

15. **Hawra F. M. A., Amal K. H.** On fuzzy Laplace transforms for fuzzy differential equations of the third order // Journal of Kerbala university. 2013. Vol. 11, N. 3. P. 251–256.

16. **Salahshour S. and Allahviranloo T.** Applications of Laplace transforms // Soft computing. 2013. Vol. 17, N. 1. P. 145–148.

17. **Najafi N., Parypour M., Lotfi T.** A new computational method for fuzzy Laplace transforms by the differential transform method // Mathematical inverse problems. 2015. Vol. 2, N. 1. P. 16–31.

18. **Stepnicka M.** Fuzzy transform and its applications to problems in engineering practice // Ph. D. Thesis. University of Ostrava, 2007. 194 с.

19. **Математические основы теории автоматического управления** / Под ред. Б. К. Чемоданова. Часть II. М.: Изд-во МГТУ им. Н. Э. Баумана, 2008. 714 с.

20. **Puri M. L., Ralesku D. A.** Differential of fuzzy functions // Journal of Mathematical analyses and applications. 1983. Vol. 91, N. 2. P. 552–558.

21. **Bede B. and Gal S. G.** Solutions of fuzzy differential equations based on generalized differentiability // Communications in Mathematical Analysis. 2010. N. 9. P. 22–41.

N. P. Demenkov, Ph. D., Associate Professor, e-mail: dnp@bmstu.ru,

E. A. Mikrin, D. Sc., Professor, e-mail: eugeny.mikrin@bmstu.ru,

I. A. Mochalov, D. Sc., Professor, e-mail: intelsyst@mail.ru,

Department of Automatic Control Systems Bauman Moscow State Technical University

Fuzzy Transformation of Laplace in Tasks of Fuzzy Mathematical Modelling. Part 1

Article is devoted to a class of models of linear and nonlinear control systems which are represented by traditional mathematical models in their fuzzy realization that provides them adequacy at different impact of indignations.

Linear and nonlinear objects of control, sensors, executive mechanisms and other components of control systems in the conditions of uncertainty lead to various types of fuzzy models: an fuzzy initial task for the linear and nonlinear differential equations with constant and variable coefficients; fuzzy one — point and two — point boundary tasks for the differential equations of the first and second orders; the fuzzy integrated equations of Voltaire like convolution of the 1 and 11 sort; the fuzzy equations in private derivatives of the 2nd order; fuzzy integro-differential equations and others.

The purpose of the real work consists in synthesis of representations of fuzzy transformation of Laplace, its properties and its application for the solution of the listed above various fuzzy linear and nonlinear tasks.

Definitions of fuzzy direct and return transformation of Laplace are given and are specified signs of his difference from traditional transformation. Types and number of the fuzzy differential equations are defined and which are solved by application of fuzzy transformation of Laplace.

The solution of these equations is closely connected with representation of a derivative in fuzzy interpretation.

The generalized fuzzy derivatives of Hukuhara are defined of various types: the fuzzy generalized type gH derivative Hukuhara, the indistinct strong generalized derivatives of $gH(i) — gH(ii)$ of types of Hukuhara. The theorem of differentiation is formulated for fuzzy derivative of the first order.

Application of fuzzy transformation of Laplace is shown for the solution of various fuzzy linear and nonlinear tasks.

Keywords: fuzzy boundary task, fuzzy transformation of Laplace, fuzzy direct and return transformation of Laplace, derivative on Hukuhara

References

1. **Методы робастного, нейро-нечеткого и адаптивного управления.** Под общ. ред. К. А. Пупкова (Methods of robustny, neuro and fuzzy and adaptive control). Moscow: Izd-vo MGTU im. N. Je. Baumana, 2001, 743 p. (in Russian).

2. **Demenkov N. P.** Nечеткое управление в технических системах (Fuzzy control in technical systems). Moscow: Izd-vo MGTU im. N. Je. Baumana, 2005, 198 p. (in Russian).

3. **Ul'janov S. V., Litvinceva L. V., Dobrynin B. N., Mishin A. A.** Интеллектуальное робастное управление: технология мягких вычислений (Intellectual robustny control: technology of soft calculations). Moscow: VNIIGeosistem, 2011, 408 p. (in Russian).

4. **Pegat A.** Nечеткое моделирование и управление (Fuzzy modeling and control). Moscow: Binom. Laboratorija znanij, 2009, 798 p. (in Russian).

5. **Volkov N. K., Kanatnikov A. A.** Integral'nye преобразования и операционное исчисление (Integrated transformations and operational calculation). Moscow: Izd-vo MGTU im. N. Je. Baumana, 2006, 227 p. (in Russian).

6. **Kolmogorov A. N., Fomin S. V.** Jelementy teorii funkcij i funkcional'nogo analiza (Elements of the theory of functions and functional analysis). Moscow: Fizmatgiz, 2006, 544 p. (in Russian).

7. **Kaleva O.** Fuzzy differential equations, *Fuzzy sets and systems*, 1987, vol. 24, no. 3, pp. 301–317.

8. **Seikkala S.** On the fuzzy initial value problem, *Fuzzy sets and systems*, 1987, vol. 24, no. 3, pp. 319–330.

9. Friedman M., Ming M., Kandel A. Fuzzy linear systems, *Fuzzy sets and systems*, 1998, vol. 96, no. 2, pp. 201–209.
10. Buckley J. J., Feuring J. Fuzzy differential equations, *Fuzzy sets and systems*, 2000, vol. 110, no. 1, pp. 43–54.
11. Mochalov I. A., Hrisat M. S., Shihab Eddin M. Ja. Nechetkie uravnenija v chastnyh proizvodnyh v zadachah upravlenija (The fuzzy differential equations in private derivatives in tasks of control). *Informacionnye tehnologii*, 2015, vo. 21, no. 8, pp. 563–569 (in Russian).
12. Hukuhara M. Integration des applications mesurables dont ia valuer est un compact convexe, *Funkcialay Ekvaciq*, 1967, no. 10, pp. 205–223.
13. Sindu Devi S. and Ganesan K. Solution of first order fuzzy differential by fuzzy Laplace transform, *Global journal of pure and applied mathematics (GJPAM)*, 2016, vol. 12, no. 4, pp. 381–387.
14. Eljaoni E., Melliani S. and Chadli S. Solving second-order fuzzy differential equations by fuzzy Laplace transform method, *Advances in difference equations?* 2015, no. 66, pp. 1–14. DOI: 10.1186/s13662-015-0414-x.
15. Hawra F. M. A., Amal K. H. On fuzzy Laplace transforms for fuzzy differential equations of the third order, *Journal of Kerbala university*, 2013, vo. 11, no. 3, pp. 251–256.
16. Salahshour S. and Allahviranloo T. Applications of Laplace transforms, *Soft computing*, 2013, vol. 17, no. 1, pp. 145–148.
17. Najafi N., Parypour M., Lotfi T. A new computational method for fuzzy Laplace transforms by the differential transform method, *Mathematical inverse problems*, 2015, vol. 2, no. 1, pp. 16–31.
18. Stepnicka M. Fuzzy transform and its applications to problems in engineering practice, *Ph. D. Thesis. University of Ostrava*, 2007, 194 p.
19. **Matematicheskie osnovy teorii avtomaticheskogo upravlenija.** Pod red. B. K. Chemojanova. Chast' II. Moscow: Izd-vo MG TU im. N. Je. Bauman, 2008. 714 p. (in Russian).
20. Puri M. L., Ralesku D. A. Differential of fuzzy functions, *Journal of Mathematical analyses and applications*, 1983, vol. 91, no. 2, pp. 552–558.
21. Bede B., Gal S. G. Solutions of fuzzy differential equations based on generalized differentiability, *Communications in Mathematical Analysis*, 2010, no. 9, pp. 22–41.

УДК 004.62

Г. С. Вересников, канд. техн. наук, ст. науч. сотр., e-mail: veresnikov@mail.ru,
Л. А. Панкова, канд. техн. наук, ст. науч. сотр., e-mail: ludmila_pankova@bk.ru,
В. А. Пронина, канд. техн. наук, ст. науч. сотр., e-mail: pron@ipu.ru,

Учреждение Российской академии наук Институт проблем управления РАН, г. Москва

Использование неопределенного программирования в проектировании

Работа посвящена применению теории неопределенности Б. Лю в проектировании. Рассматривается обобщенная модель неопределенного многокритериального программирования. Применение модели демонстрируется на примере задачи оптимального проектирования симметричной двухэлементной опорной конструкции.

Ключевые слова: теория неопределенности, неопределенное программирование, мера неопределенности, неопределенная переменная, эпистемическая неопределенность, ожидаемое значение неопределенной переменной, дисперсия неопределенной переменной, критические значения неопределенной переменной, обобщенная модель неопределенного многокритериального программирования, Парето-фронт

Введение

Предварительное проектирование сложных технических систем — длительный итерационный процесс принятия решений, предполагающий автоматизацию не только расчетной части, но и решения оптимизационных задач. Оптимизационные задачи — обычно задачи большой размерности и многокритериальные, при этом целевые функции и ограничения чаще всего нелинейные. Большинство параметров на этапе предварительного проектирования не могут быть точно определены к моменту принятия решения, т. е. являются неточными, или неопределенными. В этом случае применение традиционных методов, предназначенных для вычислений с точными значениями, может привести к недопустимым решениям [1]. Возникает проблема предварительного проектирования сложных технических систем с неточными (неопределенными) параметрами. Для разрешения этой проблемы переходят к моделям неопределенного

программирования — моделям оптимизации с неточными (неопределенными) параметрами.

Различают два типа неопределенности: алеаторную и эпистемическую. Алеаторная неопределенность возникает, когда параметры характеризуются вариабельностью, зафиксированной в результатах наблюдений (статистических данных). В этом случае параметры представляются функциями распределения вероятностей и в задачах оптимизации применяют традиционные стохастические методы.

Эпистемическая неопределенность возникает из-за недостатка знаний, результатов наблюдений (исторических или экспериментальных). В этом случае информацию получают от экспертов.

Для представления эпистемической неопределенности существует более 20 "теорий неопределенности" [2].

Наиболее популярными являются теория нечетких множеств [3], основанная на теории нечетких множеств Л. Заде, и теория возможностей [4] —

расширение теории нечетких множеств и нечеткой логики Л. Заде. Предложены различные расширения нечетких множеств, такие как: нечеткие множества второго типа (когда значениями функции принадлежности являются нечеткие множества), интервальные нечеткие множества (interval-valued fuzzy sets), нечеткие параметризованные мягкие множества (fuzzy parameterized soft sets) и др.

В 2011 г. для учета надежности предоставляемой информации о неопределенной переменной Заде ввел концепцию Z -числа как упорядоченную пару нечетких чисел (A, B) : первая компонента — значение переменной, вторая компонента — степень уверенности в первой компоненте [5]. Обычно A и B описываются в терминах естественного языка, например, около 45 мин, очень уверен.

С 2002 г. многие теории, моделирующие эпистемическую неопределенность об алеаторной неопределенности (суждения экспертов о случайном параметре), т. е. неопределенность второго порядка [6], объединены научным направлением "неточные вероятности" (imprecise probability), поддерживаемым сформировавшимся международным научным сообществом (<http://www.sipta.org/>).

Для учета эпистемической неопределенности в задачах оптимизации необходимо обеспечить:

- формальное представление неопределенности;
- метод распространения неопределенности входных неопределенных параметров на функцию от этих параметров (метод вычисления неопределенности функции от неопределенных параметров);
- модели оптимизации с неопределенными параметрами.

В нечетком [3] и возможностном [4] программировании существуют эффективные методы распространения неопределенности параметров на линейные целевые функции и функции ограничений, позволяющие свести модели нечеткого и возможностного программирования к моделям линейного программирования. Для нелинейных функций в нечетком и возможностном программировании авторам неизвестны эффективные методы распространения и решения задач оптимизации.

К настоящему времени в теории Z -чисел нет результатов, позволяющих применять их при решении практических оптимизационных задачах.

В работах по неточным вероятностям описано несколько подходов для распространения неточных вероятностей. Однако соответствующие методы требуют больших вычислительных затрат. На практике используются главным образом традиционные стохастические модели, даже когда вероятности известны неточно [7].

Теория неопределенности, предложенная Б. Лю в 2007 г. (усовершенствована в 2009 г.) [8—10], является новым инструментом для представления эпистемической неопределенности и манипулиро-

вания с ней. В неопределенном программировании теории неопределенности Б. Лю рассматривается класс задач оптимизации: целевые функции и функции ограничений включают независимые параметры с эпистемической неопределенностью и строго монотонны по этим параметрам. Для этого класса задач в теории неопределенности Б. Лю существуют эффективные методы распространения неопределенности, а модели неопределенного программирования сводятся к эквивалентным моделям математического программирования. В дальнейшем термин "неопределенность" будем использовать только в контексте теории неопределенности Б. Лю.

В разделе 1 описываются модели неопределенного программирования. Рассматривается обобщенная модель неопределенного программирования. Обсуждается выбор модели неопределенного программирования. В разделе 2 приводится методика решения задач неопределенного программирования. В разделе 3 демонстрируется применение моделей неопределенного программирования при решении задачи оптимального проектирования симметричной двухэлементной опорной конструкции [11]. В приложении даны необходимые сведения из теории неопределенности.

1. Модели неопределенного программирования

Неопределенное программирование — это математическое программирование с неопределенными параметрами: целевые функции и функции ограничений включают неопределенные параметры. В теории неопределенности [8—10] для рассматриваемого класса задач оптимизации определены аналитические выражения для ограничений и характеристик целевых функций (ожидаемое значение, дисперсия, критические значения неопределенной переменной), которые являются детерминированными дубликатами целевых функций и ограничений. Детерминированные дубликаты позволяют свести задачи неопределенного программирования к задачам математического программирования, что обеспечивает вычислительную эффективность.

Эпистемическая неопределенность в теории неопределенности — это степень уверенности эксперта в том, что событие произойдет. В основе теории неопределенности лежат три базовых понятия (см. Приложение):

- мера неопределенности $M\{\cdot\}$, удовлетворяющая аксиомам нормальности, дуальности, субаддитивности, произведения;
- вещественная неопределенная величина ξ ;
- функция распределения неопределенной величины $\Phi(x) = M\{\xi \leq x\}$.

Функции распределения неопределенных переменных строятся с помощью экспертов.

Пусть \mathbf{x} — вектор решений, ξ — вектор параметров, $f(\mathbf{x}, \xi)$ — целевая функция, $g_j(\mathbf{x}, \xi) \leq 0$, $j = 1, 2, \dots, p$ — ограничения.

Если ξ — вектор четких параметров, то задача математического программирования имеет вид:

$$\min(\max) f(\mathbf{x}, \xi) \text{ при условии} \\ g_j(\mathbf{x}, \xi) \leq 0, j = 1, 2, \dots, p.$$

Если ξ — вектор неопределенных параметров, то целевая функция $f(\mathbf{x}, \xi)$ и функции ограничений $g_j(\mathbf{x}, \xi) \leq 0, j = 1, 2, \dots, p$, при каждом фиксированном векторе \mathbf{x} являются неопределенными переменными со своими функциями распределения. Тогда, чтобы поставить задачу оптимизации, переходят к детерминированным моделям заменой целевой функции и ограничений детерминированными дубликатами.

В качестве дубликата целевой функции $d[f(\mathbf{x}, \xi)]$ используют характеристики этой неопределенной функции: ожидаемое значение $E[f(\mathbf{x}, \xi)]$ (определение 7, Приложение), дисперсия $V[f(\mathbf{x}, \xi)]$ (определение 8, Приложение), критические значения (α -оптимистическое значение $\sup_{\alpha}[f(\mathbf{x}, \xi)]$, α -пессимистическое значение $\inf_{\alpha}[f(\mathbf{x}, \xi)]$) (определение 9, Приложение).

Так как неопределенные функции $g_j(\mathbf{x}, \xi), j = 1, 2, \dots, p$, представляются функциями распределения, в случае мягких ограничений экспертом задаются доверительные уровни $\alpha_1, \alpha_2, \dots, \alpha_p, 0 \leq \alpha_j \leq 1, j = 1, 2, \dots, p$, степеней уверенности выполнения ограничений, и дубликаты ограничений будут выглядеть так:

$$M\{g_j(\mathbf{x}, \xi) \leq 0\} \geq \alpha_j, j = 1, 2, \dots, p.$$

В случае жестких ограничений $\alpha_j = 1, j = 1, 2, \dots, p$, дубликаты ограничений будут иметь вид:

$$\max\{g_j(\mathbf{x}, \xi) \leq 0\}, j = 1, 2, \dots, p.$$

Выбор дубликатов неопределенной целевой функции является прерогативой лица, принимающего решение (ЛПР), так как отражает его личные предпочтения относительно оптимизации системы. Дубликат "ожидаемое значение" соответствует "оптимизации в среднем", и его использование уместно, когда ЛПР должно принимать решения многократно.

С позиции ЛПР, не склонного к риску, чем более предсказуема ситуация, тем лучше. Решение, которое обеспечивает большую степень уверенности в отсутствие значительных отклонений от ожидаемого значения, выглядит предпочтительнее для ЛПР. В этом случае дубликатом неопределенной целевой функции может быть дисперсия.

В ряде случаев ЛПР хочет гарантировать с некоторой уверенностью, что значение целевой функции в случае минимизации не превысит соответствующего пессимистического значения, т. е. $M\{f(\mathbf{x}, \xi) \leq r\} \geq \alpha$, где α — желаемая степень уверенности, r — α -пессимистическое значение. Тогда дубликатом неопределенной целевой функции мо-

жет быть α -пессимистическое значение. В случае максимизации дубликатом неопределенной целевой функции может быть α -оптимистическое значение.

В рассматриваемом классе задач в теории неопределенности определены аналитические выражения для характеристик функции от неопределенных переменных и ограничений (теоремы 2, 3, Приложение). Таким образом, задача неопределенного программирования с использованием дубликата $d[f(\mathbf{x}, \xi)]$ неопределенной целевой функции и дубликатов ограничений становится детерминированной задачей — задачей математического программирования:

$$\min(\max) d[f(\mathbf{x}, \xi)]$$

при условии

$$M\{g_j(\mathbf{x}, \xi) \leq 0\} \geq \alpha_j, j = 1, 2, \dots, p.$$

Перейдем к задаче многокритериального математического программирования. Пусть $f_i(\mathbf{x}, \xi), i = 1, 2, \dots, m$, — целевые функции. Если ξ — вектор четких параметров, то задача многокритериального математического программирования имеет вид:

$$\min(\max)(f_1(\mathbf{x}, \xi), f_2(\mathbf{x}, \xi), \dots, f_m(\mathbf{x}, \xi))$$

при условии

$$g_j(\mathbf{x}, \xi) \leq 0, j = 1, 2, \dots, p.$$

Если ξ — вектор неопределенных параметров, то обобщенная модель многокритериального неопределенного программирования имеет вид:

$$\min(\max)(D_1[f_1(\mathbf{x}, \xi)], D_2[f_2(\mathbf{x}, \xi)], \dots, D_m[f_m(\mathbf{x}, \xi)]),$$

где

$$D_i[f_i(\mathbf{x}, \xi)] = \{d_1^i[f_i(\mathbf{x}, \xi)], \dots, d_{k_i}^i[f_i(\mathbf{x}, \xi)]\}$$

при условии

$$M\{g_j(\mathbf{x}, \xi) \leq 0\} \geq \alpha_j, \alpha_j \in [0, 1], j = 1, 2, \dots, p,$$

где

$$d_r^i \in D_i \subset D, r = 1, 2, \dots, k_i, i = 1, 2, \dots, m,$$

D — множество дубликатов.

Заметим, что для разных целевых функций могут быть выбраны различные дубликаты и для одной функции — несколько различных дубликатов. Обычно в многокритериальном стохастическом и неопределенном программировании используется один и тот же дубликат для всех целевых функций. Обобщенная модель позволяет учитывать различные предпочтения ЛПР относительно как одной целевой функции, так и разных целевых функций. Например, для одной целевой функции предпочтительно оптимизировать ожидаемое значение, а для другой — критическое.

2. Методика решения задач неопределенного программирования

С помощью эксперта строятся функции распределения для неопределенных параметров. Эксперт задает области значений неопределенных параметров и ряд значений функций распределения. При этом он отвечает на вопросы типа: "Какова уверенность в том, что данный неопределенный параметр ξ будет меньше или равен фиксированному значению x_i из области значений неопределенного параметра ξ ?"

Пусть эксперт определил степень уверенности для значений x_i , равную α_i . Получаем экспериментальные данные $(x_1, \alpha_1), (x_2, \alpha_2), \dots, (x_n, \alpha_n)$, где $x_1 < x_2 < \dots < x_n$; $0 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n \leq 1$. На основе данных, полученных от эксперта, выполняется линейная аппроксимация функции распределения каждого неопределенного параметра, которая дает так называемую эмпирическую функцию распределения [10] (рис. 1):

$$\Phi(x) = \begin{cases} 0, & \text{если } x \leq x_1; \\ \alpha_i + \frac{(\alpha_{i+1} - \alpha_i)(x - x_i)}{x_{i+1} - x_i}, & \text{если } x_i \leq x < x_{i+1}, 1 \leq i < n; \\ 1, & \text{если } x > x_n. \end{cases}$$

При аппроксимации могут использоваться другие типы функций распределения неопределенного параметра: линейная, зигзагообразная, нормальная, логнормальная и др. [10].

Опишем этапы решения задачи неопределенного программирования с целевыми функциями $f_i(\mathbf{x}, \xi)$, $i = 1, 2, \dots, m$, и ограничениями $g_j(\mathbf{x}, \xi) \leq 0$, $j = 1, 2, \dots, p$, где \mathbf{x} — вектор решения, ξ — вектор неопределенных параметров:

1. Построение с помощью эксперта функций распределения для неопределенных параметров.

2. Выбор множеств дубликатов D_i , $i = 1, 2, \dots, m$, для целевых функций.

3. Выбор метода вычисления дубликатов целевых функций и ограничений (например, вычисление интеграла при расчете ожидаемого значения методом трапеций).

5. Выбор метода оптимизации (например, эволюционный).

6. Выбор решения из найденных Парето-решений.

3. Применение обобщенной модели неопределенного многокритериального программирования

Продемонстрируем применение моделей неопределенного многокритериального программирования для решения задачи оптимального проектирования симметричной двухэлементной опорной конструкции [11]. Конструкция состоит из двух стальных труб, закрепленных вместе на одном конце, и подвергается воздействию силы f в точке, которая отклоняется на величину v (рис. 2).

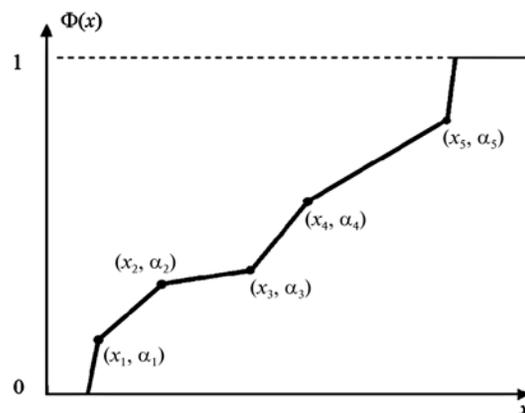


Рис. 1. Эмпирическая функция распределения

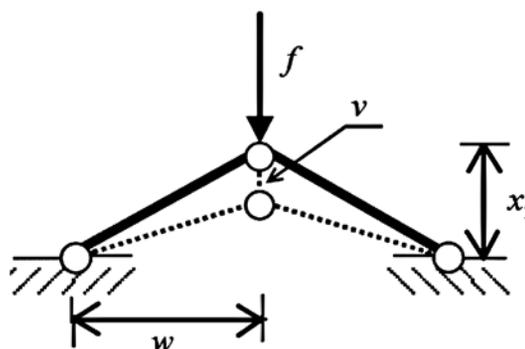


Рис. 2. Симметричная двухэлементная опорная конструкция

Оптимизируемыми функциями являются масса конструкции m и вертикальное отклонение v под нагрузкой f

$$\min(m, v),$$

где

$$m = 2\pi\rho t x_1 \sqrt{w^2 + x_2^2};$$

$$v = \frac{f(w^2 + x_2^2)^{3/2}}{2\pi t E x_1 x_2^2},$$

при условии, что давление s на трубы должно быть меньше критического s_{cr} :

$$s \leq s_{cr}$$

где

$$s = \frac{f}{2\pi t x_1 x_2} \sqrt{w^2 + x_2^2};$$

$$s_{cr} = \frac{1}{8} \pi^2 E \frac{t^2 + x_1^2}{w^2 + x_2^2},$$

$$1 \leq x_1 \leq 100, 10 \leq x_2 \leq 1000.$$

Модуль упругости E , толщина труб t , плотность массы ρ — константы ($E = 210 \text{ кН} \cdot \text{мм}^2$, $t = 2,5 \text{ мм}$, $\rho = 7,8 \cdot 10^{-3} \text{ г/мм}^3$). Диаметр труб x_1 и высота кон-

струкции x_2 — проектируемые переменные. Неопределенными параметрами являются нагрузка f (кН) и ширина конструкции w (мм): $\xi = (f, m)$.

Рассмотрим эту задачу как задачу неопределенного программирования.

Неопределенные параметры f и w задаются интервалами: $120 \leq f \leq 180$, $650 \leq w \leq 850$. Будем считать, что эксперт задал для неопределенных параметров линейные функции распределения $L(a, b)$, где a и b — границы: $L_f(120, 180)$, $L_w(650, 850)$. Линейная функция распределения неопределенности представлена ниже:

$$\Phi(x) = \begin{cases} 0, & \text{если } x \leq a; \\ (x - a)/(b - a), & \text{если } a \leq x \leq b; \\ 1, & \text{если } x \geq b. \end{cases}$$

Обратная функция имеет вид:

$$\Phi^{-1}(\alpha) = (1 - \alpha)a + \alpha b, 0 \leq \alpha \leq 1.$$

Пусть ЛПР выбрал дубликаты: для массы конструкции m — ожидаемое значение, для отклонения v — α -пессимистическое значение при $\alpha = 0,95$. Ищется решение $\mathbf{x} = (x_1, x_2)$ такое, что:

- ожидаемое значение массы конструкции $E[m]$ минимально по \mathbf{x} ;
- отклонение v не больше $\inf_{0,95}[v]$ с уверенностью 0,95 и минимально по \mathbf{x} .

Такой выбор дубликатов говорит о разном отношении ЛПР к целевым функциям: отклонение — критический параметр с точки зрения ЛПР, в отличие от массы конструкции.

Тогда постановка задачи проектирования симметричной двухэлементной опорной конструкции как задачи многокритериального неопределенного программирования с мягкими ограничениями имеет вид:

$$\min(E[m], \inf_{0,95}[v])$$

при условии $M\{s - s_{cr} \leq 0\} \geq 0,95$, $1 \leq x_1 \leq 100$, $10 \leq x_2 \leq 1000$, $120 \leq f \leq 180$, $650 \leq w \leq 850$.

Так как целевые функции и функции ограничения — строго возрастающие относительно независимых переменных f и w , задача неопределенного

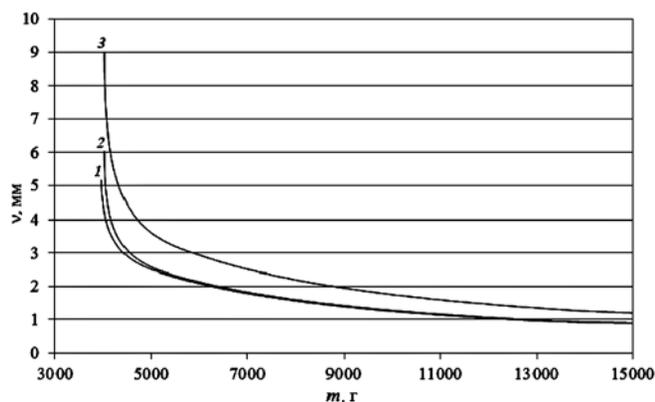


Рис. 3. Аппроксимация Парето-фронт, полученных при $120 \leq f \leq 180$, $650 \leq w \leq 850$

программирования сводится к точной постановке с использованием аналитических выражений для $E[m]$, $\inf_{0,95}[v]$ и $M\{s - s_{cr} \leq 0\} \geq 0,95$:

$$\begin{aligned} \min_{x_1, x_2} E[m] &= \\ &= \min_{x_1, x_2} \int_0^1 2\pi r t x_1 \sqrt{((1 - \alpha) \cdot 650 + \alpha \cdot 850)^2 + x_2^2} d\alpha, \end{aligned}$$

$$\begin{aligned} \min_{x_1, x_2} \inf_{0,95}[v] &= \\ &= \min_{x_1, x_2} \frac{(0,05 \cdot 120 + 0,95 \cdot 180)((0,05 \cdot 650 + 0,95 \cdot 850)^2 + x_2^2)^{3/2}}{2\pi t E x_1 x_2^2} \end{aligned}$$

при условии

$$\frac{0,05 \cdot 120 + 0,95 \cdot 180}{2\pi t x_1 x_2} \sqrt{(0,05 \cdot 650 + 0,95 \cdot 850)^2 + x_2^2} -$$

$$- \frac{1}{8} \pi^2 E \frac{t^2 + x_1^2}{(0,05 \cdot 650 + 0,95 \cdot 850)^2 + x_2^2} \leq 0;$$

$$1 \leq x_1 \leq 100, 10 \leq x_2 \leq 1000,$$

$$120 \leq f \leq 180, 650 \leq w \leq 850.$$

Для решения поставленной задачи применялся многокритериальный генетический алгоритм пакета Матлаб 2012 (solver Multiobjective optimization using Genetic Algorithm). На рис. 3 показан результат двухкритериальной оптимизации в виде аппроксимации Парето¹-фронт:

- для модели со средними значениями неопределенных переменных (кривая 1);
- для модели неопределенного многокритериального программирования с дубликатами: ожидаемое значение для массы конструкции m и отклонения v (кривая 2);
- для модели неопределенного многокритериального программирования с дубликатами: ожидаемое значение для массы конструкции m и 0,95 — пессимистическое значение для отклонения v (кривая 3).

Если неопределенные параметры f и w задаются функциями неопределенности на интервалах: $90 \leq f \leq 210$, $500 \leq w \leq 1000$, то результат оптимизации в виде аппроксимации Парето-фронт приобретает вид, показанный на рис. 4.

Можно видеть, что при расширении интервала изменения неопределенных переменных увеличивается отличие в решениях, полученных при использовании рассмотренных моделей.

В табл. 1 даны фрагменты Парето-фронт и соответствующие Парето-решения для кривой 1,

¹ Парето-решение — недоминируемое по Парето решение. Недоминируемость по Парето — невозможность улучшения любой оптимизируемой функции без ухудшения других. Парето-множество — множество недоминируемых по Парето решений. Парето-фронт — образ Парето-множества в пространстве оптимизируемых функций.

в табл. 2 — для кривой 2, в табл. 3 — для кривой 3 (рис. 4).

Как видно из табл. 1–3, если применять модель с точными параметрами, то не гарантируется сохранение выбранных по кривой 1 значений массы

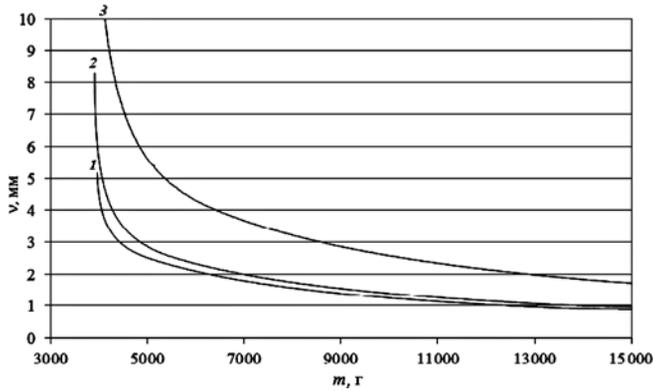


Рис. 4. Аппроксимация Парето-фронт, полученных при $90 \leq f \leq 210$, $500 \leq w \leq 1000$

Таблица 1

Парето-фронт 1 и Парето-решения

№ п/п	m	v	x_1	x_2
1	4511,173	2,888097	37,93987	615,8729
2	4523,093	2,882031	38,06115	615,0272
3	4528,124	2,882005	38,14628	613,3100
4	4529,461	2,871250	38,02154	618,7699
5	4533,415	2,870971	38,08601	617,5146
6	4535,967	2,858552	37,95161	623,7717
7	4547,217	2,852965	38,06785	622,8849
8	4552,460	2,842455	38,00189	627,2870

Таблица 2

Парето-фронт 2 и Парето-решения

№ п/п	$E[m]$	$E[v]$	x_1	x_2
1	4507,11	3,464521	39,01255	564,6638
2	4511,561	3,459266	39,03958	565,1204
3	4518,399	3,447923	39,06012	566,6529
4	4531,572	3,427942	39,10968	569,1984
5	4531,587	3,427931	39,10981	569,1984
6	4537,185	3,411939	39,08031	572,2759
7	4541,662	3,404768	39,09324	573,2891
8	4556,124	3,389936	39,19024	574,3724

Таблица 3

Парето-фронт 3 и Парето-решения

№ п/п	$E[m]$	$\inf_{0,95}[v]$	x_1	x_2
1	4511,984	7,090994	39,02632	565,7926
2	4529,955	7,048756	39,14915	567,0831
3	4535,965	6,985780	39,07429	572,0983
4	4544,692	6,984475	39,17856	570,9493
5	4546,90	6,940969	39,10033	574,7901
6	4546,928	6,940929	39,10057	574,7899
7	4569,726	6,854275	39,16555	579,9447
8	4569,754	6,854230	39,16580	579,9449

m и отклонения v из-за вариации неопределенных параметров, особенно в области малых значений m (при больших значениях массы m отклонение v менее чувствительно к изменениям неопределенных параметров f и w). Кроме того, решение, выбранное по детерминированной модели, может оказаться недопустимым.

Из табл. 1–3 видно, что для того чтобы гарантировать выбранное отклонение v , следует обеспечить большее значение массы (по кривой 3), чем соответствующее этому отклонению значение массы на кривой 2. Таким образом, принятие решений с использованием $E[m]$ и $E[v]$ может быть неудовлетворительным для ЛПР, не склонного к риску. Альтернативой может стать использование $E[m]$ и $\inf[v]$.

Заключение

С использованием теории неопределенности Б. Лю был эффективно решен ряд практических задач оптимизации в различных областях: управление запасами [11], управление портфелем проектов [12], размещение пожарных станций [13], логистики поставок зерна [14]. Работа посвящена применению теории неопределенности Б. Лю в проектировании, где необходимы учет неопределенности параметров и эффективность вычислений. Рассматривается обобщенная модель неопределенного программирования при эпистемической неопределенности, моделируемой мерой неопределенности Б. Лю. Приведено решение задачи оптимального проектирования симметричной двухэлементной опорной конструкции как задачи неопределенного программирования. Показана целесообразность применения обобщенной модели, позволяющей реализовывать различные предпочтения ЛПР относительно оптимизации системы. Актуальной задачей для дальнейших исследований является разработка робастных моделей неопределенного программирования.

Приложение

Из теории неопределенности [8–10]. Пусть U — непустое универсальное множество, A — σ -алгебра на U . Каждое $\Lambda \in A$ называется событием. Каждому событию Λ ставится в соответствие число $M\{\Lambda\}$, определяющее меру неопределенности этого события, удовлетворяющее трем аксиомам.

Аксиома нормальности: $M\{U\} = 1$.

Аксиома дуальности: $M\{\Lambda\} + M\{\Lambda^c\} = 1$ ($\Lambda^c = U \setminus \Lambda$).

Аксиома субаддитивности: для каждой счетной последовательности событий

$$M\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \leq \sum_{i=1}^{\infty} M\{\Lambda_i\}.$$

Определение П1. Пусть U — непустое универсальное множество, A — σ -алгебра на U , M — мера неопределенности. Тогда триплет (U, A, M) называется измеримым **пространством неопределенности**.

Аксиома произведения: пусть (U_k, A_k, M_k) — пространства неопределенности для $k = 1, 2, \dots$, тогда мера неопределенности на произведении σ -алгебр $A_1 \times A_2 \times \dots$ удовлетворяет равенству

$$M\left\{\prod_{k=1}^{\infty} \Lambda_k\right\} = \bigwedge_{k=1}^{\infty} M\{\Lambda_k\}, \text{ где } \bigwedge \triangleq \inf.$$

Определение П2. Неопределенная переменная есть измеримая функция ξ на пространстве неопределенности (U, A, M) в множество действительных чисел, т. е. для любого борелевского множества B действительных чисел множество $\{\xi \in B\} = \{u \in U | \xi(u) \in B\}$ есть событие.

Определение П3. Функция распределения неопределенности неопределенной переменной ξ есть функция $\Phi: R \rightarrow [0, 1]$, определяемая так:

$$\Phi(x) = M\{\xi \leq x\}.$$

Определение П4. Функция $\Phi(x)$ распределения неопределенной переменной ξ называется **регулярной**, если $\Phi(x)$ непрерывная строго возрастающая по x , $0 < \Phi(x) < 1$ и $\lim_{x \rightarrow -\infty} \Phi(x) = 0$, $\lim_{x \rightarrow +\infty} \Phi(x) = 1$.

Определение П5. Пусть ξ — неопределенная переменная с регулярной функцией распределения $\Phi(x)$. Тогда обратная функция $\Phi^{-1}(\alpha)$ называется **обратным распределением** ξ .

Определение П6. Неопределенные переменные $\xi_1, \xi_2, \dots, \xi_n$ называются **независимыми**, если

$$M\left\{\bigcup_{i=1}^{\infty} (\xi_i \in B_i)\right\} = \bigwedge_{k=1}^{\infty} M(\xi_k \in B_k)$$

для любых борелевских множеств B_1, B_2, \dots, B_n .

Неопределенные переменные независимы, если они определены на разных пространствах неопределенности.

Теорема 1. Пусть $\xi_1, \xi_2, \dots, \xi_n$ — неопределенные переменные, f — вещественная измеримая функция. Тогда $f(\xi_1, \xi_2, \dots, \xi_n)$ является неопределенной переменной.

Определение П7. Ожидаемое значение неопределенной переменной ξ определяется как среднее значение неопределенной переменной в смысле неопределенной меры:

$$E[\xi] = \int_0^{+\infty} M\{\xi \geq r\} dr - \int_{-\infty}^0 M\{\xi \leq r\} dr.$$

Определение П8. Дисперсия неопределенной переменной ξ определяется как среднее значение квад-

рата отклонения неопределенной переменной от ее ожидаемого значения в смысле неопределенной меры:

$$V[\xi] = E(\xi - E[\xi])^2.$$

Теорема 2. Пусть функция $f(\xi_1, \xi_2, \dots, \xi_n)$ — непрерывная строго возрастающая по $\xi_1, \xi_2, \dots, \xi_m$ и строго убывающая по $\xi_{m+1}, \xi_{m+2}, \dots, \xi_n$. Тогда, если $\xi_1, \xi_2, \dots, \xi_n$ — независимые неопределенные переменные с регулярными функциями распределения неопределенности $\Phi_1, \Phi_2, \dots, \Phi_n$ соответственно, то:

a) $\xi = f(\xi_1, \xi_2, \dots, \xi_n)$ является неопределенной переменной с обратным распределением неопределенности:

$$\Psi^{-1}(\xi) = f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)),$$

$$b) E[\xi] = \int f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)) d\alpha,$$

$$c) V[\xi] = \int_0^1 (f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)) - E[\xi])^2 d\alpha,$$

$$d) \text{ для любого } \alpha \in [0, 1] \\ M\{f(\xi_1, \xi_2, \dots, \xi_n) \leq 0\} \geq \alpha$$

эквивалентно

$$f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_m^{-1}(\alpha), \Phi_{m+1}^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)) \leq 0.$$

Определение П9. Критические значения неопределенной переменной ξ определяются как

$$\sup_{\alpha} [\xi] = \sup\{r | M\{\xi \geq r\} \geq \alpha\} - \alpha\text{-оптимистическое значение,}$$

$$\inf_{\alpha} [\xi] = \inf\{r | M\{\xi \leq r\} \geq \alpha\} - \alpha\text{-пессимистическое значение,}$$

где $\alpha \in [0, 1]$.

Теорема 3. Пусть $f(\xi_1, \xi_2, \dots, \xi_n)$ — непрерывная строго возрастающая по $\xi_1, \xi_2, \dots, \xi_n$. Тогда, если $\xi_1, \xi_2, \dots, \xi_n$ — независимые неопределенные переменные с регулярными распределениями $\Phi_1, \Phi_2, \dots, \Phi_n$ соответственно, то

a) если $\xi = f(\xi_1, \xi_2, \dots, \xi_n)$ — непрерывная строго возрастающая по $\xi_1, \xi_2, \dots, \xi_n$, то

$$\sup_{\alpha} [\xi] = f(\Phi_1^{-1}(1-\alpha), \Phi_2^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)),$$

$$\inf_{\alpha} [\xi] = f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_n^{-1}(\alpha));$$

b) если $\xi = f(\xi_1, \xi_2, \dots, \xi_n)$ — непрерывная строго убывающая по $\xi_1, \xi_2, \dots, \xi_n$, то

$$\sup_{\alpha} [\xi] = f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \dots, \Phi_n^{-1}(\alpha)),$$

$$\inf_{\alpha} [\xi] = f(\Phi_1^{-1}(1 - \alpha), \Phi_2^{-1}(1 - \alpha), \dots, \Phi_n^{-1}(1 - \alpha)).$$

Список литературы

1. **Nemirovski A.** Lectures on Robust Convex Optimization // URL: http://www2.isye.gatech.edu/~nemirovs/RO_LN.pdf (accessed 1.06.2016).
2. **Zimmerman H.-J.** Fuzzy set theory // Inc. WIREs Comp Stat. John Wiley & Sons. 2010. Vol. 2. P. 317–332.
3. **Zimmerman H.-J.** Fuzzy Set Theory and Applications. 4th Rev. ed. Boston: Kluwer Academic Publishers, 2001. 514 p.
4. **Wagenknecht M., Yazenin A.** Possibilistic optimization. Tver', TvGU, 2012.
5. **Zadeh L. A.** A Note on Z-numbers // Information Sciences. 2011. N. 181. P. 2923–2932.
6. **Zadeh L.** Computation with imprecise probabilities // URL: <http://www.gimac.uma.es/ipmu08/proceedings/papers/zadeh.pdf> (дата обращения 1.06.2016).
7. **Bruns M., Paredisa C., Ferson S.** Computational Methods for Decision Making based on Imprecise Information // Proceedings of the Reliable Engineering Computing Workshop (REC'06), Savannah, GA, February. 2006. P. 22–24.
8. **Liu B.** Uncertainty Theory. 2-rd edition. Berlin: Springer-Verlag. 2007. 485 p.
9. **Liu B.** Theory and Practice of Uncertain Programming. 3-rd edition. Berlin: Springer-Verlag. 2009. 201 p.
10. **Liu B.** Uncertainty Theory. URL: <http://orsc.edu.cn/liu/ut.pdf> (дата обращения 1.06.2016).
11. **Messac A., Ismail-Yahaya A.** Multiobjective robust design using physical programming // Structural and Multidisciplinary Optimization. 2002. Vol. 23, N. 5. P. 357–371.
12. **Lixia Rong.** Two New Uncertainty Programming Models of Inventory with Uncertain Costs // Journal of Information & Computational Science. 2011. Vol. 8, N. 2. P. 280–288.
13. **Rupak Bhattacharyya, Amitava Chatterjee, Samarjit Kar.** Uncertainty Theory Based Novel Multi-Objective Optimization Technique Using Embedding Theorem with Application to R & D Project Portfolio Selection // Applied Mathematics. 2010. Vol. 1. P. 189–199.
14. **Jian Zhou, Zhen Li, Ke Wang.** A Multi-Objective Model for Fire Station Location under Uncertainty // Advances in Information Sciences and Service Sciences. 2013. Vol. 5, N. 7. P. 1184–1191.
15. **Sibo Ding.** A New Uncertain Programming Model for Grain Supply Chain Design // Information: An International Interdisciplinary Journal. 2013. Vol. 16, N. 2 (A). P. 1069–1076.

G. S. Veresnikov, Senior Researcher, e-mail: veresnikov@mail.ru,

L. A. Pankova, Senior Researcher, e-mail: ludmila_pankova@bk.ru,

V. A. Pronina, Senior Researcher, e-mail: pron@ipu.ru,

Institute of Control Sciences V. A. Trapeznikov Academy of Sciences, Moscow

Use of Uncertain Programming in Design

The paper is devoted to application of the uncertainty theory offered by B. Liu in design. The generalized model of uncertain multiobjective programming is considered. The use of the model is demonstrated on the two-bar truss design optimization.

Keywords: *uncertainty theory, uncertain programming, uncertainty measure, uncertain variable, epistemic uncertainty, expected value of uncertain variable, variance of uncertain variable, critical values of uncertain variable, generalized model of uncertain multiobjective programming, Pareto-front*

References

1. **Nemirovski A.** Lectures on Robust Convex Optimization. URL: http://www2.isye.gatech.edu/~nemirovs/RO_LN.pdf (accessed 1.06.2016).
2. **Zimmerman H.-J.** Fuzzy set theory, Inc. WIREs Comp Stat. John Wiley & Sons, 2010, vol. 2, pp. 317–332.
3. **Zimmerman H.-J.** Fuzzy Set Theory and Applications. 4th Rev. ed. Boston: Kluwer Academic Publishers, 2001. 514 p.
4. **Wagenknecht M., Yazenin A.** Possibilistic optimization. Tver', TvGU, 2012.
5. **Zadeh L. A.** A Note on Z-numbers, *Information Sciences*, 2011, no. 181, pp. 2923–2932.
6. **Zadeh L.** Computation with imprecise probabilities // URL: <http://www.gimac.uma.es/ipmu08/proceedings/papers/zadeh.pdf> (accessed 1.06.2016).
7. **Bruns M., Paredisa C., Ferson S.** Computational Methods for Decision Making based on Imprecise Information, *Proceedings of the Reliable Engineering Computing Workshop (REC'06)*, Savannah, GA, February. 2006, pp. 22–24.
8. **Liu B.** *Uncertainty Theory*. 2-rd edition. Berlin: Springer-Verlag, 2007, 485 p.
9. **Liu B.** *Theory and Practice of Uncertain Programming*. 3-rd edition. Berlin: Springer-Verlag, 2009, 201 p.
10. **Liu B.** *Uncertainty Theory*. URL: <http://orsc.edu.cn/liu/ut.pdf> (accessed 1.06.2016).
11. **Messac A., Ismail-Yahaya A.** Multiobjective robust design using physical programming, *Structural and Multidisciplinary Optimization*, 2002, vol. 23, no. 5, pp. 357–371.
12. **Lixia Rong.** Two New Uncertainty Programming Models of Inventory with Uncertain Costs, *Journal of Information & Computational Science*, 2011, vol. 8, no. 2, pp. 280–288.
13. **Rupak Bhattacharyya, Amitava Chatterjee, Samarjit Kar.** Uncertainty Theory Based Novel Multi-Objective Optimization Technique Using Embedding Theorem with Application to R & D Project Portfolio Selection, *Applied Mathematics*, 2010, vol. 1, pp. 189–199.
14. **Jian Zhou, Zhen Li, Ke Wang.** A Multi-Objective Model for Fire Station Location under Uncertainty, *Advances in Information Sciences and Service Sciences*, 2013, vol. 5, no. 7, pp. 1184–1191.
15. **Sibo Ding.** A New Uncertain Programming Model for Grain Supply Chain Design, *Information: An International Interdisciplinary Journal*, 2013, vol. 16, no. 2 (A), pp. 1069–1076.

УДК 004.032.26

В. И. Горбаченко, д-р техн. наук, проф., зав. каф., e-mail: gorvi@mail.ru,
С. Н. Катков, ст. преподаватель, e-mail: skat.pnz@yandex.ru,
Пензенский государственный университет, г. Пенза

Мемристорные клеточные сети для решения краевых задач¹

Показана возможность использования мемристоров в аналоговых клеточных сетях для решения краевых задач. Вычислительную систему, содержащую мемристорную сеть, предлагается реализовать в виде двухслойной архитектуры. Предлагается алгоритм аддитивной коррекции итерационного решения сеточных уравнений на клеточных сетях, обеспечивающий точность решения, ограниченную только цифровой частью вычислительной системы.

Ключевые слова: клеточная сеть, мемристор, алгоритм аддитивной коррекции, сходимость, устойчивость

Введение

До появления мощных цифровых компьютеров основным средством решения краевых задач, описываемых дифференциальными уравнениями в частных производных (ДУЧП), были аналоговые сеточные модели. Идея использования сетки из сопротивлений для решения уравнения Лапласа принадлежит С. А. Гершгорину [1]. Аналоговое сеточное моделирование прошло долгий путь от простейших моделей до весьма мощных для своего времени аналого-цифровых вычислительных комплексов [2–5]. Достоинством сеточных моделей является высокий параллелизм: теоретически решение получается мгновенно (практически — после завершения переходного процесса, обусловленного паразитными параметрами сети). Так как решение формируется за счет аналоговых связей между всеми элементами сети, то деградация и отказ отдельных элементов приводит только к снижению точности решения, а не к потере решения. Кроме того, аналоговые компьютеры основаны на явлении аналогии и всегда позволяют получить физически правдоподобное решение плохо обусловленных задач, решение которых на цифровых компьютерах требует больших затрат времени или вообще не может быть получено.

Недостатками сеточных моделей, общими для аналоговых вычислительных машин, которые привели к переходу от аналоговой к цифровой технике, являлись низкая технологичность элементной базы, недостаточная точность решения и неуниверсальный характер аналоговых компьютеров.

В настоящее время в области решения краевых задач, описываемых ДУЧП, в определенной степе-

ни "преемниками" сеточных моделей являются аналоговые линейные клеточные сети — один из видов клеточных нейронных сетей (CNN — Cellular Neural Networks) [6–10].

Цель данной работы — анализ возможностей решения систем разностных уравнений, аппроксимирующих краевые задачи, на клеточных сетях, построенных на новых элементах — мемристорах. Предлагаются пути преодоления основного недостатка аналоговых моделей — недостаточной точности решения.

Аналоговые линейные клеточные сети

Аналоговая линейная клеточная сеть (далее — клеточная сеть) — это система простых процессоров (cell — ячеек), расположенных на регулярной сетке и связанных между собой в одном или более слоях.

Теоретически можно определить клеточную сеть любого измерения и с произвольным шаблоном, но для простоты ограничимся двумерным случаем. Пример двумерной клеточной сети, использующей пятиточечный шаблон, показан на рис. 1. Любая ячейка в клеточной сети связана только с соседни-

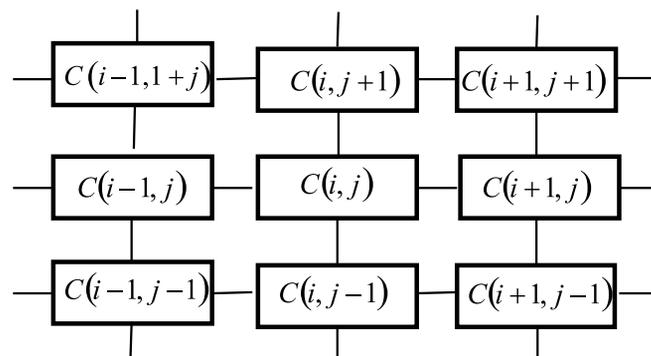


Рис. 1. Пример структуры клеточной сети

¹ Работа поддержана грантами РФФИ 16-08-00906, 14-01-00660 и 14-01-00733.

ми ячейками в соответствии с шаблоном. Шаблон ячейки $C(i, j)$ — это множество (окрестность) $N(i, j)$ ячеек $C(k, l)$, с которыми связана ячейка $C(i, j)$, и веса связи. Вес связи определяется значениями коэффициентов связи (параметрами шаблона) между выходами ячеек, входящих в шаблон, и входами ячейки $C(i, j)$. В шаблон также могут входить внешние входы, коэффициенты управления, смещение и другие элементы.

Уравнение состояния ячейки клеточной сети с непрерывным представлением времени имеет вид [8, 10]

$$c_{ij} \frac{du_{ij}(t)}{dt} = d_{ij} u_{ij}(t) + \sum_{C(k, l) \in N(i, j)} g_A(i, j; k, l) u_{kl}(t), \quad (1)$$

где c_{ij} — числовой коэффициент (в электрической схеме ячейки [10] — входная емкость); $u_{ij}(t)$ — переменная состояния ячейки; d_{ij} — числовой коэффициент (в электрической схеме ячейки [10] — входная проводимость ячейки); $u_{kl}(t)$ — выходы ячеек, входящих во множество $N(i, j)$ ячеек шаблона; $g_A(i, j; k, l)$ — весовой коэффициент (вес связи), соединяющий ячейки $C(i, j)$ и $C(k, l)$.

Относительно параметров сети приняты следующие предположения:

$$g_A(i, j; k, l) = g_A(k, l; i, j), \quad d_{ij} > 0, \quad c_{ij} > 0.$$

Состояния и веса являются непрерывными значениями, сама сеть является линейной.

В клеточных сетях с дискретным представлением времени [8–11] производная по времени в уравнении (1) представляется соотношением

$$\frac{du_{ij}(t)}{dt} = \frac{u_{ij}(t) - u_{ij}(t-1)}{\Delta t},$$

где Δt — шаг по времени.

Тогда с учетом сделанных предположений уравнение (1) можно записать в следующем виде:

$$\left[-\frac{c_{ij}}{\Delta t} + d_{ij} \right] u_{ij}(t) + \sum_{C(k, l) \in N(i, j)} g_A(i, j; k, l) u_{kl}(t) = -\frac{c_{ij}}{\Delta t} u_{ij}(t-1). \quad (2)$$

В клеточной сети с дискретным представлением времени состояния сети изменяются в дискретные моменты времени.

Решение на клеточной сети систем разностных уравнений

Система алгебраических уравнений (2) связана масштабными коэффициентами с системой разностных уравнений, аппроксимирующей по неявной схеме краевую задачу с различными граничными

условиями. Действительно, пусть в некоторой двумерной области Ω решается ДУЧП

$$k \frac{\partial w}{\partial t} = Lw, \quad (3)$$

где L — линейный дифференциальный оператор.

На границе области Ω заданы граничные условия и начальные условия при $t = 0$. Заменяя входящие в Lw (3) производные разностными отношениями, получим разностное выражение $L_h w_h$, являющееся линейной комбинацией значений сеточной функции w_h на некотором множестве $M(x_{ij})$ узлов сетки — сеточном шаблоне:

$$(L_h w_h)_{ij} = \sum_{x_{kl} \in M(x_{ij})} a_h(i, j; k, l) w_h(x_{kl}),$$

где $a_h(i, j; k, l)$ — коэффициенты, значения которых зависят от конкретной формы разностной аппроксимации; h — шаг сетки; $M(x_{ij})$ — шаблон в узле x_{ij} . Такую же форму имеет аппроксимация граничных условий. Соотношения справедливы для произвольной размерности области решения. Применяя неявную аппроксимацию производной по времени в (3), получаем

$$k_{ij} \frac{w_h(x_{ij}, t) - w_h(x_{ij}, t-1)}{\tau} = \sum_{x_{kl} \in M(x_{ij})} a_h(i, j; k, l) w_h(x_{kl}, t),$$

или

$$\left[-\frac{k_{ij}}{\tau} + a_h(i, j; i, j) \right] w_h(x_{ij}, t) + \sum_{\substack{x_{kl} \in M(x_{ij}) \\ x_{kl} \neq x_{ij}}} a_h(i, j; k, l) w_h(x_{kl}, t) = -\frac{k_{ij}}{\tau} w_h(x_{ij}, t-1). \quad (4)$$

В выражении (4) суммирование проводится по "окрестности" узла x_{ij} . При условии равенства шаблонов клеточной сети и разностной схемы выражения (2) и (4) подобны, если связать параметры разностной схемы и клеточной сети масштабными соотношениями

$$g_A(i, j; k, l) = m_g a_h(i, j; k, l); \quad g_f(i, j) = \frac{c_{ij}}{\Delta t} = m_g \frac{k_{ij}}{\tau};$$

$$I(i, j) = \frac{c_{ij}}{\Delta t} u_{x_{ij}}(t-1) = m_i \frac{k}{\tau} w_h(x_{ij}, t-1); \quad (5)$$

$$U(i, j, t) = m_u w_h(x_{ij}, t);$$

$$U(i, j, t-1) = m_u w_h(x_{ij}, t-1).$$

Практически удобно реализовать связи ячейки с соседними ячейками с помощью пассивных электрических проводимостей. Это возможно, если коэффициенты $a_h(i, j; i, j)$ и $a_h(i, j; k, l)$ имеют разные знаки при $k \neq i$ и $l \neq j$ и выполняется условие

диагонального преобладания $|a_h(i, j; i, j)| \geq \sum_{\substack{x_{kl} \in M(x_{ij}) \\ x_{kl} \neq x_{ij}}} a_h(i, j; k, l)$, что справедливо для боль-

шинства разностных схем. Тогда выход клеточной сети связан с сеточной функцией соотношением $m_u w_h(x_{ij}, t) = u_{ij}(t)$. Причем должен выполняться индикатор подобия $m_g m_u / m_i = 1$. Веса связи моделируются электрическими проводимостями $g_A(i, j; k, l)$, правая часть уравнения (4) может моделироваться током $I(i, j)$ или напряжением $U(i, j, t-1)$, подаваемым через проводимость $g_i(i, j)$, выход сети моделируется напряжением $U(i, j, t)$. Отсюда m_g , m_i и m_u — соответственно, масштабы по проводимости, току и напряжению.

Так как переход к сети с дискретным представлением времени основан на неявной схеме аппроксимации, то клеточная сеть с дискретным представлением времени является абсолютно устойчивой при любых значениях шагов по пространству и времени. Клеточная сеть с дискретным представлением времени проще сети с непрерывным представлением времени, позволяет использовать итерационную настройку параметров при решении нелинейных задач, и, как будет показано ниже, позволяет уточнять на цифровых процессорах решение, полученное на аналоговых сетях.

Таким образом, клеточная сеть для решения ДУЧП — это сеть управляемых проводимостей, источников напряжения и тока, топология ее совпадает с топологией разностной сетки, а параметры связаны масштабными соотношениями с параметрами разностной схемы. Для двумерной сети (см. рис. 1) возможные варианты ячеек, построенных по выражениям (5), показаны на рис. 2 и 3. Схемы

рис. 2, 3 электрически эквивалентны, но схема рис. 3, как будет показано ниже, позволяет уточнять решение, полученное на сети. Поэтому далее будем рассматривать схему рис. 3. Приведенные результаты легко распространяются на случай стационарных задач.

Рассмотренную сеть целесообразно реализовать в составе аналого-цифровой вычислительной системы. В цифровой части системы выполняется расчет конечно-разностной аппроксимации и параметров клеточной сети. Система задания параметров сети задает параметры проводимостей, источников напряжения и тока. Для этого управляемые проводимости, источники напряжения и тока должны представлять собой цифроаналоговые преобразователи. После задержки, обусловленной переходным процессом за счет паразитных емкостей сети, система съема решения измеряет и преобразует в цифровую форму узловые напряжения сети. В цифровой части системы после обратного масштабирования формируется решение задачи.

Особенностью такой системы является то, что для большинства задач решение может быть получено за одну итерацию в отличие от существующих цифровых систем. Работа систем задания и съема решения легко может быть распараллелена путем обслуживания отдельных областей сети. Современная элементная база позволяет строить точные и компактные цифроаналоговые преобразователи "код—напряжение", "код—ток" и аналого-цифровые преобразователи. Точные и компактные преобразователи "код—проводимость" реализовать на традиционной элементной базе весьма сложно. Поэтому перспективным направлением реализации клеточных сетей является использование нового элемента электронных схем — мемристора. Мемристоры в перспективе позволят реализовать ком-

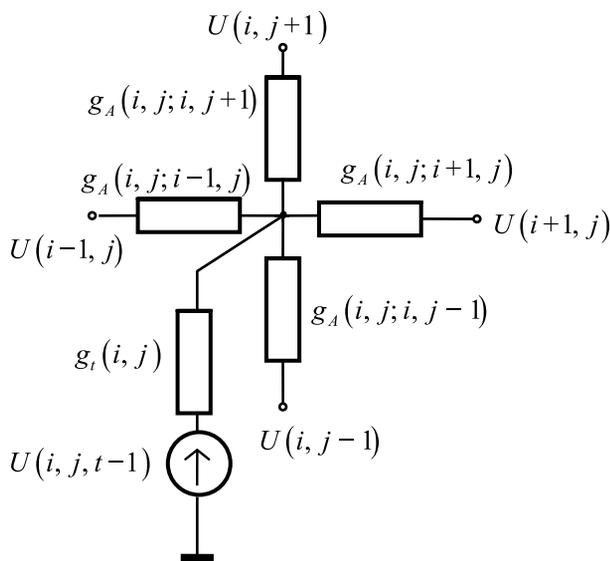


Рис. 2. Схема ячейки клеточной сети с управляемым источником напряжения

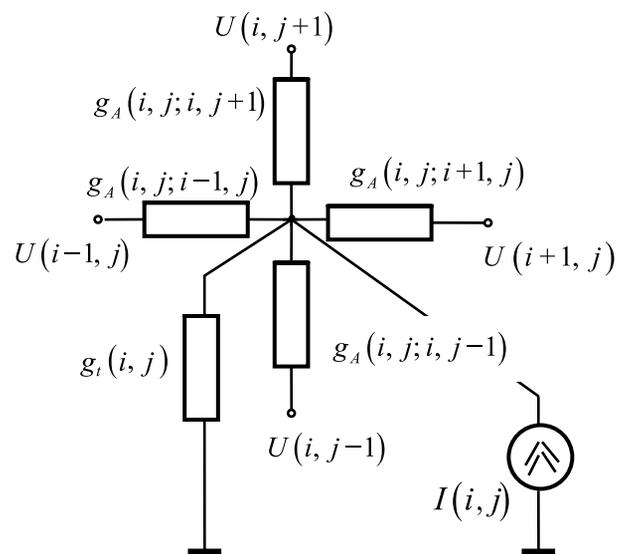


Рис. 3. Схема ячейки клеточной сети с управляемым источником тока

пактные и точные преобразователи "код—проводимость". А так как мемристор является наноэлектронным прибором, то его применение позволит минимизировать задержку, вызванную переходными процессами в паразитных емкостях, и качественно сократить время решения задачи.

Мемристорные клеточные сети

Еще в 1971 г. Л. Чуа теоретически предложил новый двухэлектродный элемент, названный "мемристор" [12]. Но только в 2008 г. фирма *Hewlett-Packard* реализовала мемристор в виде микросхемы. Л. Чуа выдвинул и математически обосновал гипотезу о том, что наряду с индуктивностью, конденсатором и резистором должен быть четвертый базовый элемент электрических цепей. Л. Чуа исходил из того, что должны быть соотношения, связывающие все четыре основные переменные электрических цепей: ток i , напряжение u , заряд q и магнитный поток Φ . Всего таких соотношений может быть шесть. Пять из них хорошо известны:

$$R(i) = \frac{du(i)}{di}, \quad C(q) = \frac{dq(u)}{du}, \quad L(\Phi) = \frac{d\Phi(i)}{di},$$

$$i(t) = \frac{dq(t)}{dt}, \quad u(t) = \frac{d\Phi(t)}{dt},$$

где $u(t)$ и $i(t)$ — переменные напряжение и ток; $q(t)$ и $\Phi(t)$ — заряд и магнитный поток; $R(i)$, $C(q)$ и $L(\Phi)$ — сопротивление, емкость и индуктивность, зависящие, соответственно, от значений тока, заряда и магнитного потока.

Л. Чуа предположил, что должно существовать шестое соотношение, связывающее магнитный поток с зарядом:

$$M(q) = \frac{d\Phi(q)}{dq},$$

откуда $\frac{d\Phi(t)}{dt} = M(q) \frac{dq}{dt}$, или $u(t) = M(q)i(t)$. Так как

$q(t) = \int_{-\infty}^t i(\tau) d\tau$, то недостающий элемент — мемристор

тор описывается выражением

$$u(t) = M\left(\int_{-\infty}^t i(\tau) d\tau\right) i(t),$$

где M называется мемрезистивностью (общепринятого перевода нет, англ. *memristance*).

Мемрезистивность зависит от тока. Мемристор является нелинейным элементом с памятью. Современные мемристоры реализуются средствами нанoeлектроники [13, 14]. Мемристор можно рассматривать как управляемый резистор. Причем он может работать и как цифровой элемент памяти, находящийся в одном из двух состояний (с малым или высоким сопротивлением), и как управляемый резистор. Установка требуемых значений проводимости мемристоров может быть реализована путем подачи на мемристор импульса напряжения заданных значения и длительности [14]. Очень важно, что состояние мемристора сохраняется при отключении питания. Поэтому мемристоры перспективны в качестве запоминающих и логических элементов и управляемых резисторов. В частности, мемристоры как переменные резисторы очень перспективны для реализации синаптических весов искусственных нейронов [14—18]. Разработаны различные схемы реализации синаптических весов с использованием мемристоров [14—20]. В работе [20] предложена реализация на мемристорах весов клеточных нейронных сетей.

Использование мемристоров в клеточных сетях означает возврат к аналоговым методам обработки информации, но на принципиально иной технологической основе. Мемристоры, являющиеся изделиями нанoeлектроники, решают проблему нетехнологичности и громоздкости старых аналоговых машин. Использование нейронных блоков как второго слоя в архитектуре аналого-цифрового суперкомпьютера [21] снимает проблему неуниверсальности и, как будет показано ниже, недостаточной точности аналогового компьютера.

Веса связи клеточной сети (g_A и g_I в соотношениях (5)) могут быть реализованы с помощью мемристоров [22]. Согласно работе [23] мемристоры можно разделить на два класса — "цифровые" и "аналоговые". В "цифровых" мемристорах сопротивление практически не меняется, если напряжение на мемристоре не превышает некоторого порога, зависящего от материала мемристора. При превышении порога сопротивление быстро изменяется (за наносекунды) и значением этого изменения можно управлять, подбирая длительность и амплитуду импульса напряжения. В "аналоговых" мемристорах изменение сопротивления является непрерывной функцией приложенного смещения. Таким образом, "цифровые" мемристоры можно использовать при постоянных напряжениях, если они не превышают порога. Сопротивление мемристоров можно регулировать, подавая импульсы амплитудой V выше или ниже пороговых значений $V_{\text{порог}}$ и $-V_{\text{порог}}$, зависящих от материала мемристора, длительностью τ . Длительность импульса рассчитывают исходя из знания максимальной G_{max} и минимальной G_{min} проводимостей мемристоров, которые сильно различаются в зависимости от материала мемристора. Синаптические веса сети g_A и g_I в проводимости мемристоров отображаются с помощью масштабного соотношения [14]

$$g_A(i, j; k, l) = \frac{a_h(i, j; k, l) - a_{h\text{min}}}{a_{h\text{max}} - a_{h\text{min}}} (G_{\text{max}} - G_{\text{min}}) + G_{\text{min}},$$

где $a_{h\text{min}} \leq a_h(i, j; k, l) \leq a_{h\text{max}}$, аналогично вычисляют величины $g_I(i, j)$.

Тогда клеточная сеть представляет собой сеть из мемристоров, реализующих веса g_A и соединенных в соответствии с шаблоном сети. Узлы сети соединены мемристорами, реализующими g_p с общей точкой ("землей") сети. К узлам сети должны быть подключены схемы задания тока $I(i, j)$ (см. рис. 3) и схемы измерения узловых напряжений. Мемристоры могут также выполнять функции аналоговой памяти и коммутирующих элементов. Это позволяет построить аналоговую клеточную сеть на современной нанoeлектронной базе. Применение мемристоров позволит качественно снизить время решения задачи как за счет сокращения времени настройки проводимостей, так и за счет малой длительности переходного процесса в нанoeлектронных схемах. Сеть должна быть также снабжена системой измерения решения в производных ячейках, системами подготовки данных, хранения и обработки решения. Таким образом, архитектура вычислительной системы, включающей клеточную мемристорную сеть, должна быть двухслойной [21]: первый (внутренний) слой — мемристорная сеть, второй (внешний) слой — система параллельно работающих процессоров подготовки данных и обработки результатов. Второй слой должен иметь архитектуру и реализацию, обеспечивающие высокоскоростную обработку данных. В противном случае преимущества мемристорной реализации клеточной сети не будут реализованы. Во втором слое также перспективным является применение мемристоров, например, в качестве запоминающих элементов.

Алгоритм аддитивной коррекции

Рассмотренная клеточная сеть реализует метод пряминой аналогии. Точность решения, получаемого на сети, определяется точностью аналоговых компонентов. Считается, что аналоговая техника обеспечивает невысокую точность решения. Однако следует учитывать, что исходные данные задачи, как правило, задаются приближенно. Поэтому решение задачи до малых значений формальных показателей точности (например, нормы невязки) не всегда оправдано. Для плохо обусловленных задач, каковыми являются многие реальные задачи, малые значения формальных показателей точности не гарантируют действительную точность решения. Поэтому для многих задач предлагаемые сети могут обеспечить приемлемую точность решения. Следует также отметить их устойчивость по отношению к ошибкам в исходных данных.

Погрешности клеточной сети можно компенсировать, используя алгоритм аддитивной коррекции [8]. Отличительной особенностью алгоритма является то, что на сети формируется только поправка к решению, а само решение вычисляется в цифровой части системы. При сходимости процесса обучения и его устойчивости к аппаратным погрешностям алгоритм аддитивной коррекции обеспечивает

точность, принципиально ограниченную лишь возможностями цифровой части системы. Как показал опыт использования алгоритма аддитивной коррекции на сеточных моделях [8], для получения высокой точности решения достаточно нескольких итераций, тогда как решение на цифровых вычислительных системах требует очень большого числа итераций.

Для описания алгоритма аддитивной коррекции используем матричное описание сети. Система разностных уравнений может быть представлена в матричной форме в виде системы линейных алгебраических уравнений $\mathbf{Aw} = \mathbf{b}$. Если последовательно пронумеровать ячейки сети, то система (2) может быть записана в виде системы линейных алгебраических уравнений: $\mathbf{Gu} = \mathbf{i}$, причем ввиду погрешности сети $\mathbf{G} \approx m_g \mathbf{A}$ и при принятых допущениях матрица \mathbf{G} является симметричной положительно определенной. Матрица \mathbf{G} является матрицей проводимостей связи межклеточных соединений, вектор $\mathbf{i} = m_p \mathbf{b}$ — вектор токов, подаваемых в узлы сети, вектор $\mathbf{u} = m_u \mathbf{w}$ — вектор напряжений выхода сети. Фактически сеть моделирует не матрицу \mathbf{A} , а положительно определенную матрицу $\mathbf{B} = 1/m_g \mathbf{G}$.

Алгоритм аддитивной коррекции представляет собой итерационный процесс решения системы $\mathbf{Aw} = \mathbf{b}$. Начальным приближением является решение, полученное на сети и преобразованное в цифровую форму. На k -м шаге алгоритма в цифровой части двухслойной системы рассчитывается вектор невязки $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{Aw}^{(k-1)}$ и масштаб по току $m_i^{(k)} = I_{\max} / \|\mathbf{r}^{(k)}\|$, где I_{\max} — максимальный по абсолютному значению ток, который можно задавать в сети, $\|\mathbf{r}^{(k)}\| = \max_i |r_i|$ — кубическая норма вектора невязки. В методе аддитивной коррекции принципиально важно использовать весь диапазон изменения токов смещения. Рассчитывается вектор токов $\mathbf{i}^{(k)} = m_i^{(k)} \mathbf{r}^{(k)}$ и задается в сеть. На моделирующей сети формируется вектор напряжений поправки к решению, который формально можно описать матричным равенством $\Delta \mathbf{u}^{(k)} = \mathbf{G}^{-1} \mathbf{i}^{(k)}$. Вектор $\Delta \mathbf{u}^{(k)}$ преобразуется в цифровую форму. В цифровой части вычислительной системы после обратного масштабирования вычисляется вектор поправки к решению $\Delta \mathbf{w}^{(k)} = (m_u^{(k)})^{-1} \Delta \mathbf{u}^{(k)} = \mathbf{B}^{-1} \mathbf{r}^{(k)}$ и формируется новое приближение решения $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} + \omega_k \Delta \mathbf{w}^{(k)}$, где ω_k — подбираемый или вычисляемый, например, одним из методов спуска, итерационный параметр. Рассчитывается невязка полученного приближения и т. д. Процесс продолжается до тех пор, пока не будет получено малое значение нормы невязки. Рассмотренный алгоритм можно трактовать как предобусловленный итерационный процесс [24], а клеточную сеть — как аппаратно реализованный предобусловливатель.

Получим условия сходимости алгоритма аддитивной коррекции при $\omega_k = \omega = \text{const} > 0$ и положительно определенной матрице \mathbf{B} . Приближение решения на k -й итерации описывается выражением

$$\begin{aligned} \mathbf{w}^{(k)} &= \mathbf{w}^{(k-1)} + \omega \Delta \mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} + \omega \mathbf{B}^{-1} \mathbf{r}^{(k)} = \\ &= \mathbf{w}^{(k-1)} + \omega \mathbf{B}^{-1} (\mathbf{b} - \mathbf{A} \mathbf{w}^{(k-1)}) = \\ &= (\mathbf{E} - \omega \mathbf{B}^{-1} \mathbf{A}) \mathbf{w}^{(k-1)} + \omega \mathbf{B}^{-1} \mathbf{b} = \mathbf{C} \mathbf{w}^{(k-1)} + \mathbf{d}, \end{aligned} \quad (6)$$

где \mathbf{E} — единичная матрица, $\mathbf{d} = \mathbf{B}^{-1} \mathbf{b}$,

$$\mathbf{C} = \mathbf{E} - \omega \mathbf{B}^{-1} \mathbf{A}, \quad (7)$$

где \mathbf{C} — итерирующая матрица.

Необходимое и достаточное условие сходимости итерационного процесса (6) имеет вид $\rho(\mathbf{C}) < 1$, где ρ — спектральный радиус матрицы (максимальное по абсолютной величине собственное значение матрицы). Для оценки спектрального радиуса рассмотрим задачу на собственные значения для итерирующей матрицы $\mathbf{C} \mathbf{x} = \lambda \mathbf{x}$, где \mathbf{x} — произвольный ненулевой вектор, λ — собственное значение. С учетом (7) приходим к обобщенной задаче на собственные значения:

$$(\mathbf{B} - \omega \mathbf{A}) \mathbf{x} = \lambda \mathbf{B} \mathbf{x}. \quad (8)$$

Все собственные значения в (8) будут вещественными, поскольку матрица $\mathbf{B} - \omega \mathbf{A}$ является самосопряженной как линейная комбинация матриц \mathbf{A} и \mathbf{B} , а матрица \mathbf{B} положительно определена. Используя обобщенное отношение Рэлея, запишем условие сходимости в виде

$$-1 < \frac{((\mathbf{B} - \omega \mathbf{A}) \mathbf{x}, \mathbf{x})}{(\mathbf{B} \mathbf{x}, \mathbf{x})} < 1, \quad (9)$$

где $((\mathbf{B} - \omega \mathbf{A}) \mathbf{x}, \mathbf{x})$ и $(\mathbf{B} \mathbf{x}, \mathbf{x})$ — скалярные произведения.

С учетом положительной определенности матрицы \mathbf{B} выражение (9) имеет вид

$$0 < \omega (\mathbf{A} \mathbf{x}, \mathbf{x}) < 2(\mathbf{B} - \omega \mathbf{A}),$$

что эквивалентно неравенствам

$$(\mathbf{A} \mathbf{x}, \mathbf{x}) > 0, \quad (10)$$

$$((2\mathbf{B} - \omega \mathbf{A}) \mathbf{x}, \mathbf{x}) > 0. \quad (11)$$

Условие (10) означает положительную определенность матрицы разностных уравнений. Выполнение условия (11) обеспечивается выбором масштаба по проводимости и параметра ω . При одинаковых шаблонах разностной схемы и клеточной сети достаточное условие сходимости имеет вид

$$\begin{aligned} g_A(i, j; k, l) &> 0,5 \omega m_g |a_h(i, j; k, l)|, \\ g_A(i, j) &> 0,5 \omega m_g \left[-\frac{kj}{\tau} + a_h(i, j; i, j) \right]. \end{aligned}$$

Важно, что вследствие близости матрицы преобразователя \mathbf{B} к матрице \mathbf{A} для уточнения решения требуется небольшое число итераций [8].

Устойчивость алгоритма аддитивной коррекции к погрешностям задания параметров клеточной се-

ти и измерения вектора состояния сети обеспечиваются применением переменного масштаба по току, использующего весь диапазон задания токов в сети (см. [8]).

Рассматривая клеточную сеть как аппаратно реализованный преобусловливатель, можно построить различные итерационные методы, в том числе методы подпространств Крылова [24–26], например метод сопряженных градиентов. В некоторых задачах матрица \mathbf{A} системы разностных уравнений является несимметричной и знаконеопределенной. В этом случае можно рекомендовать использовать один из методов подпространств Крылова — преобусловленный устойчивый алгоритм бисопряженных градиентов (Preconditioned BiCGSTAB) [24–26].

Для решения нелинейных задач можно использовать итерационный алгоритм с пересчетом нелинейных параметров задачи на каждом временном шаге. На каждой итерации на сети решается задача со своими параметрами. Причем можно не изменять параметры сети, а использовать аддитивную коррекцию.

Если размеры сеточной области превышают размеры клеточной сети, то можно использовать блочные, в том числе асинхронные блочные методы [24].

Заключение

Таким образом, решение краевых задач, описываемых дифференциальными уравнениями в частных производных, возможно на аналоговых клеточных мемристорных сетях с дискретным представлением времени. Реализация вычислительной системы, содержащей мемристорную сеть, целесообразна в виде двухслойной архитектуры. Нижний слой представляет собой мемристорную сеть, верхний слой — параллельную цифровую систему. Использование алгоритма аддитивной коррекции позволяет получить на такой системе решение, точность которого ограничена только возможностями цифровой части системы.

Список литературы

1. Гершгорин С. А. Об электрических сетках для приближенного решения дифференциального уравнения Лапласа // Журнал прикладной физики. 1929. Т. 6, вып. 3–4. С. 3–29.
2. Гутенмахер Л. И. Электрические модели // Электричество. 1948. № 8. С. 34–40.
3. Liebmann G. Solution of Partial Differential Equations with a Resistance Network Analogue // British Journal of Applied Physics. 1950, Vol. 1, N. 4. P. 92–103.
4. Козлов Э. С., Сергеев Н. П., Николаев Н. С. Автоматизация процессов решения краевых задач. М.: Энергия, 1974. 112 с.
5. Горбаченко В. И., Катков С. Н., Козлов Э. С., Мирошкин В. А. Архитектура и алгоритмы работы сеточного мультипроцессора // Автоматизация процессов обработки первичной информации: сборник науч. трудов. Пенза: Пенз. политехн. ин-т, 1985. С. 30–40.
6. Chua L., Yang L. Cellular Neural Networks: Applications // IEEE Transactions on Circuits and Systems. 1988. Vol. 35. N. 10. P. 1273–1290.
7. Gilli M., Roska T., Chua L. O., Civalieri P. P. On the relationship between CNNs and PDEs // Proceedings of the 7th IEEE

workshop on Cellular Neural Networks and their applications, 2002. P. 16–24.

8. **Горбаченко В. И.** Нейрокомпьютеры в решении краевых задач теории поля. М.: Радиотехника, 2003. 336 с.

9. **Slavova A.** Cellular Neural Networks: Dynamics and Modeling. Springer, 2010. 220 p.

10. **Chua L., Yang L.** Cellular Neural Networks: Theory // IEEE Transactions on Circuits and Systems. 1988. Vol. 35, N. 10. P. 1257–1272.

11. **Itoh M., Chua L.** Memristor Cellular Automata and Memristor Discrete-Time Cellular Neural Networks / Memristor Networks. Springer, 2014. P. 649–713.

12. **Chua L. O.** Memristor — The Missing Circuit Element // IEEE Transactions on Circuits Theory. 1971. Vol. 18, N. 5. P. 507–519.

13. **Елисеев Н.** Мемристоры и кроссбары. Нанотехнологии для процессоров // Электроника: Наука, Технология, Бизнес. 2010. № 8. С. 84–89.

14. **Tarkov M. S.** Mapping Weight Matrix of a Neural Network's Layer onto Memristor Crossbar // Optical Memory and Neural Networks (Information Optics). 2015. Vol. 24, N. 2. P. 109–115.

15. **Галушкин А. И.** Мемристоры в развитии высокопроизводительной вычислительной техники // Информационные технологии. 2015. Т. 21, № 2. С. 146–155.

16. **Johnsen G. K.** An introduction to the memristor — a valuable circuit element in bioelectricity and bioimpedance // Journal of Electrical Bioimpedance. 2012. Vol. 3. P. 20–28.

17. **Thomas A.** Memristor-based neural networks // Journal of Physics D: Applied Physics. 2013. Vol. 46. N. 9. [Электронный ресурс]. URL <http://iopscience.iop.org/article/10.1088/0022-3727/46/9/093001/pdf;jsessionid=2D114EB9C5F3D9084B943975AЕСА3FD2.c3.iopscience.cld.iop.org> (дата обращения: 01.11.2016).

18. **Ермолаев В. А.** Вопросы применения пассивных элементов с памятью в электронных системах и нейронных сетях // Методы и устройства передачи и обработки информации. Муром: Муромский институт им. В. К. Зворыкина, 2012. Вып. 14. С. 5–10.

19. **Adhikari S. P., Yana C., Kim H., Chua L. O.** Memristor Bridge Synapse-Based Neural Network and Its Learning // IEEE Transactions on Neural Networks and Learning Systems. 2012. Vol. 23, N. 9. P. 1426–1435.

20. **Kim Y.-S., Min K.-S.** Synaptic Weighting Circuits for Cellular Neural Networks // 13th International Workshop on Cellular Nanoscale Networks and Their Applications, 29–31 August 2012. P. 1–6.

21. **Галушкин А. И.** На пути к нейрокомпьютерам с использованием мемристоров // XVI Всероссийская научно-техническая конференция с международным участием "Нейроинформатика—2014": лекции по нейроинформатике. М.: НИЯУ МИФИ, 2014. С. 62–105.

22. **Laiho M., Lehtonen L.** Memristive analog arithmetic within cellular arrays // 2012 IEEE International Symposium on Circuits and Systems (ISCAS). Seoul, 2012. P. 2665–2668.

23. **Lu W., Kim K.-H., Chang T., Gaba S.** Two-terminal resistive switches (memristors) for memory and logic applications // 16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011), Yokohama, 25–28 Jan. 2011. P. 217–223.

24. **Горбаченко В. И.** Вычислительная линейная алгебра с примерами на MATLAB. СПб.: БХВ-Петербург, 2011. 320 с.

25. **Саад Ю.** Итерационные методы для разреженных линейных систем. М.: Изд-во МГУ, 2013. 344 с.

26. **Vorst van der H.** Iterative Krylov Methods for Large Linear Systems. Cambridge: Cambridge University Press, 2003. 232 p.

V. I. Gorbachenko, D. Sc., Professor, Head of Chair, e-mail: gorvi@mail.ru,

S. N. Katkov, Senior Lecturer, e-mail: skat.pnz@yandex.ru,

Penza State University

Memristor Cellular Network for Solving Boundary Value Problems

Before the advent of high-performance digital computers primary mean of solving boundary value problems described by differential equations in partial derivatives (PDEs), was some analog simulators especially the network model. The advantages of network models are high concurrency, low sensitivity to failures of individual elements of the simulation environment and the possibility of solving ill-conditioned problems. The main drawback of the network model is the lack of solution accuracy.

Currently in the field of solving boundary value problems in a certain degree of network models "successors" are linear analog cellular network — a type of cellular neural networks.

The purpose of this paper is to analyze the possibilities of solving boundary value problems in the analog cellular networks based on new elements — memristor. The ways are offered to overcome the main shortcoming of analog models — the lack of solution accuracy.

There's given the analysis of linear analog cellular network with a discrete time representation. It's the seen connection between the mathematical description of the cellular network and the system of difference equations approximating the implicit scheme for the boundary value problem. There are presented schemes of analog cellular network cells.

It's shown the possibility of realization the cellular communication network weights using memristor. Application memristor will reduce the time to solve the problem by reducing conductivities setup time, and due to the short duration of the transition process in nanoelectronic circuits. Computing system architecture comprising memristor cellular network must be a two-layer: the first layer is a memristor network, the second layer is the system of parallel processors and data preparation processing results.

To compensate the analog cellular network error it's proposed the additive correction algorithm. In this algorithm on the net it's just formed an amendment to the decision and the decision itself is calculated in the digital part of the system that provides the accuracy of the solution, in general only limited with opportunities of the digital system. We prove the convergence and stability of the algorithm.

Keywords: boundary value problem, partial differential equation, analog cellular network, memristor, additive correction algorithm, convergence, stability

References

1. **Gershgorin S. A.** Ob jelektricheskikh setkah dlja priblizhennogo reshenija differencial'nogo uravnenija Laplasy (About the electrical networks for the approximate solution of the differential Laplace equation), *Zhurnal prikladnoj fiziki*, 1929, vol. 6. no. 3–4, pp. 3–29 (in Russian).

2. **Gutenmaher L. I.** Jelektricheskie modeli (Electric models), *Jelektrichestvo*, 1948, no. 8, pp. 34–40 (in Russian).

3. **Liebmann G.** Solution of Partial Differential Equations with a Resistance Network Analogue, *British Journal of Applied Physics*, 1950, vol. 1. no 4, pp. 92–103.

4. **Kozlov Je. S., Sergeev N. P., Nikolaev N. S.** *Avtomatizacija processov reshenija kraevykh zadach* (The automation of the boundary

value problems solutions process), Moscow, Jenergija, 1974, 112 p. (in Russian).

5. **Gorbachenko V. I., Katkov S. N., Kozlov Je. S., Miroshkin V. A.** Arhitektura i algoritmy raboty setochnogo mul'tiprocessora (The architecture and algorithms for multi-processor grid), *Avtomatizacija processov obrabotki pervichnoj informacii*, Penza, Penzenskij politehnicheskij institut, 1985, pp. 30–40 (in Russian).

6. **Chua L., Yang L.** Cellular Neural Networks: Applications, *IEEE Transactions on Circuits and Systems*, 1988, vol. 35, no. 10, pp. 1273–1290.

7. **Gilli M., Roska T., Chua L. O., Civalleri P. P.** On the relationship between CNNs and PDEs, *Proceedings of the 7th IEEE workshop on Cellular Neural Networks and their applications*, 2002, pp. 16–24.

8. **Gorbachenko V. I.** *Nejrokomputery v reshenii kraevykh zadach teorii polja* (Neurocomputers in boundary value problems solving of field theory), Moscow, Radiotekhnika, 2003, 336 p. (in Russian).

9. **Slavova A.** *Cellular Neural Networks: Dynamics and Modelling*, Springer, 2010. 220 p.

10. **Chua L., Yang L.** Cellular Neural Networks: Theory, *IEEE Transactions on Circuits and Systems*, 1988, vol. 35, no. 10, pp. 1257–1272.

11. **Itoh M., Chua L.** Memristor Cellular Automata and Memristor Discrete-Time Cellular Neural Networks, *Memristor Networks*, Springer, 2014, pp. 649–713.

12. **Chua L. O.** Memristor — The Missing Circuit Element, *IEEE Transactions on Circuits Theory*, 1971, vol. 18, no. 5, pp. 507–519.

13. **Eliseev N.** Memristory i krossbary. Nanotehnologii dlja processorov (Memristors and crossbars. Nanotechnology for processors), *Jelektronika: Nauka, Tehnologija, Biznes*, 2010, no. 8, pp. 84–89 (in Russian).

14. **Tarkov M. S.** Mapping Weight Matrix of a Neural Network's Layer onto Memristor Crossbar, *Optical Memory and Neural Networks* (Information Optics), 2015, vol. 24, no. 2, pp. 109–115.

15. **Galushkin A. I.** Memristory v razvitanii vysokoproizvoditel'noj vychislitel'noj tehniky (Memristors in the development of high-performance computing), *Informacionnye tehnologii*, 2015, vol. 21, no. 2, pp. 146–155 (in Russian).

16. **Johnsen G. K.** An introduction to the memristor — a valuable circuit element in bioelectricity and bioimpedance, *Journal of Electrical Bioimpedance*, 2012, vol. 3, pp. 20–28.

17. **Thomas A.** Memristor-based neural networks, *Journal of Physics D: Applied Physics*, 2013, vol. 46, no. 9, URL <http://iopscience.iop.org/article/10.1088/0022-3727/46/9/093001/pdf;jsessionid=2D114EB9C5F3D9084B943975AECA3FD2.c3.iopscience.cld.iop.org>

18. **Ermolaev V. A.** Voprosy primeneniya passivnykh jelementov s pamjat'ju v jelektronnykh sistemah i nejronnykh setjah (Questions of passive elements applying with the memory in electronic systems and neural networks), *Metody i ustrojstva peredachi i obrabotki informacii. Murom: Muromskij institut imeni V. K. Zvorykina*, 2012, no. 14, pp. 5–10 (in Russian).

19. **Adhikari S. P., Yana C., Kim H., Chua L. O.** Memristor Bridge Synapse-Based Neural Network and Its Learning, *IEEE Transactions on Neural Networks and Learning Systems*, 2012, vol. 23, no. 9, pp. 1426–1435.

20. **Kim Y.-S., Min K.-S.** Synaptic Weighting Circuits for Cellular Neural Networks, *13th International Workshop on Cellular Nanoscale Networks and Their Applications*, 29–31 August 2012, pp. 1–6.

21. **Galushkin A. I.** Na puti k nejrokomputeram s ispol'zovaniem memristorov (On the way to the neurocomputers using memristors), *XVI Vserossijskaja nauchno-tehnicheskaja konferencija s mezhdunarodnym uchastiem "Nejroinformatika-2014": lekcii po nejroinformatike*, Moscow, NIJaU MIFI, 2014, pp. 62–105 (in Russian).

22. **Laiho M., Lehtonen L.** Memristive analog arithmetic within cellular arrays, *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*. Seoul, 2012, pp. 2665–2668.

23. **Lu W., Kim K.-H., Chang T., Gaba S.** Two-terminal resistive switches (memristors) for memory and logic applications, *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*, Yokohama, 25–28 Jan. 2011, pp. 217–223.

24. **Gorbachenko V. I.** *Vychislitel'naja linejnaja algebra s primerami na MATLAB* (Computational linear algebra with examples in MATLAB), Sankt-Peterburg, BHV Peterburg, 2011. 320 p. (in Russian).

25. **Saad Ju.** *Iteracionnye metody dlja razrezhennykh linejnykh system* (Iterative methods for sparse linear systems). Moscow, Izdatel'stvo MGU, 2013. 344 p. (in Russian).

26. **Vorst van der H.** *Iterative Krylov Methods for Large Linear Systems*, Cambridge, Cambridge University Press, 2003, 232 p.

УДК 004.942

А. Н. Родионов, д-р техн. наук, вед. науч. сотр.
Вычислительный центр ДВО РАН, г. Хабаровск, ran@newmail.ru

Некоторые синтаксические и семантические ограничения сценарных моделей бизнес-процессов

Основная цель настоящей работы — получить простой и наглядный метод базовой проверки на корректность (бездефектность) сетей потоков работ (WF-сетей), опираясь на который можно решать не только задачи менеджмента сетей, но и специальные задачи концептуального моделирования данных, являющихся частью процессно-ориентированных информационных систем. Формально структурная модель потока работ представляет собой ориентированный граф, содержащий вершины (узлы) как минимум двух видов: узлы-действия и узлы-логические условия. Предполагается, что построенная модель процесса должна обеспечивать достижение некоторой входной последовательности фактов, возникающих по завершении определенных действий. Базовая проверка на бездефектность включает в себя проверку синтаксической и семантической корректности сети. Сеть считается синтаксически корректной, если выполняются все функциональные ограничения, исключая появление "нежелательных" связей между узлами сети. Семантическая корректность сети означает, что в графе не должны присутствовать логические узлы, являющиеся тупиками. Для проверки семантической корректности сети в процессе ее построения разрабатывается метод, использующий преобразование графа и элементы алгебры логики.

Ключевые слова: сеть потоков работ, верификация сети, бездефектность, бизнес-процессы, синтаксическая и семантическая корректность

Корректные модели бизнес-процессов — первооснова не только эффективного менеджмента, но и инженерии данных. Нахождение временных точек актуализации, исправление ретроспективных ошибок и несоответствий в записях баз данных, получение адаптивных схем организации данных на основе, например, использования паттернов и антипаттернов — вот далеко неполный перечень актуальных проблем даталогического моделирования, которые могут быть решены только при условии учета фактора потоков работ [1, 2].

Среди множества сценарных моделей процессов две их разновидности — IDEF3 и eEPC считаются наиболее востребованными. Оба класса по сути предназначены для одного и того же — детального описания последовательности шагов, необходимых для получения в общем случае некоторой предварительно заданной последовательности фактов. Оба класса содержат одинаковый набор элементов — действий и логических условий, дополненный в случае с eEPC событиями.

Главный вопрос, на который должна ответить любая построенная сценарная модель, — корректен или некорректен лежащий в ее основе реальный процесс. Считается, что процесс корректен (правдоподобен, бездефектен), если его завершение гарантировано. Для этого необходимо, чтобы в процессе отсутствовали нетерминальные связи (связи, не входящие ни в какие структурные элементы процесса), тупики, а также бесконечные циклы, вызванные тем, что условия выхода из цикла никогда не наступят [3].

Для верификации таких сетей разработано множество методов, использующих различные формальные техники, в том числе такие, как сети Петри [3–5], логика высказываний [6], специализированный язык моделирования YWAL [7]. Все они с разной степенью полноты и эффективности решают задачи корректности WF-сетей. Здесь и далее термин WF-сеть будет применяться ко всем классам сценарных моделей (а не только, как это принято, к специализированным сетям Петри), предназначенных для описания потоков работ.

В настоящей работе разрабатывается метод, который использует функциональные зависимости и элементы алгебры логики для верификаций сценарных моделей бизнес-процессов, построенных посредством таких языков, как eEPC и IDEF3. Метод не требует предварительного преобразования eEPC и IDEF3 моделей в сети Петри и может обеспечить проверку на бездефектность этих моделей в процессе их непосредственного построения. Область применения метода не распространяется на проверку корректности "нелокальной" семантики таких моделей, обусловленной использованием OR-соединений [7–9].

Выполнение любой осмысленной и завершенной последовательности действий должно преследовать определенную цель. Если руководствоваться здравым смыслом, то таковой будет достижение некоторой взаимосвязанной последовательности результатов. В контексте данных это будет означать, что в результате выполнения процесса должен быть получен набор фактов (или несколько альтернативных наборов), ассоциирующихся с объектами и образующих предметную область. Это факты, которые показывают, какие объекты появились в предметной области или покинули ее, и факты, отражающие новые взаимодействия, установившиеся между объектами. Нет никаких препятствий включить в стандартную сценарную модель ссылки на подобные факты, привязав последние к действиям, поскольку своим появлением факты обязаны именно выполнению действий.

Существуют разные точки зрения на корректность моделей бизнес-процессов. В нашем случае корректными будут считаться те модели, которые гарантируют достижение некоторой заданной последовательности фактов. (Что, однако, не отрицает необходимости в выполнении всех других вышеперечисленных условий, которые, как принято считать [3–5], обеспечивают правдоподобие бизнес-сети.)

Формально любой процесс представляет собой ориентированный граф, который содержит как минимум два типа узлов: действий (работ, функций, активностей) и логических условий (коннекторов, соединений). Посредством последних моделируется альтернативность в выполнении действий. Используемые логические операции (в нотации ARIS, например, именуемые как "логическое И (AND, \wedge)", "включающее ИЛИ (OR, \vee)" и "исключающее ИЛИ (XOR, \wp)" по своему назначению идентичны соответствующим булевым функциям: конъюнкции, строгой дизъюнкции и дизъюнкции. Следует заметить, что в IDEF3 моделях, в отличие от eEPC, проводится различие между синхронными и асинхронными логическими AND и OR. Но поскольку фактор "синхронности" действий не оказывает влияния на логику процесса, в расчет он приниматься не будет.

Присутствие логических коннекторов в сценарных моделях делает неоднозначным достижение исходной последовательности фактов. Чтобы показать это, воспользуемся фрагментом сценарной модели некоторого абстрактного бизнес-процесса (рис. 1).

Бизнес-процесс содержит набор действий a_1, a_2, \dots, a_{16} и два типа логических узлов: разворачивающих — $\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4$, иницирующих параллельное или избирательное выполнение отдельных функций, и сворачивающих — $\underline{c}_1, \underline{c}_2, \underline{c}_3, \underline{c}_4$, предполагающих, что для продолжения процесса необ-

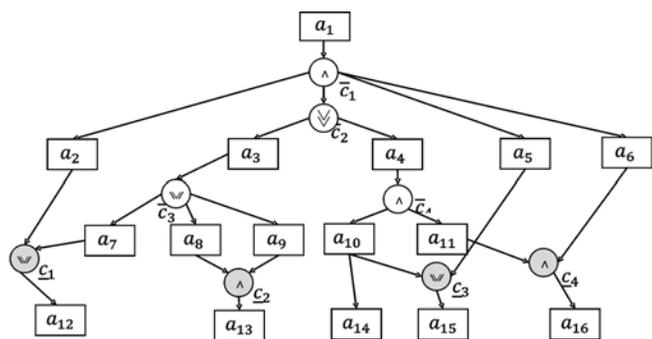


Рис. 1. Фрагмент сценарной модели бизнес-процесса

ходимо, чтобы какая-то совокупность предшествующих действий завершилась.

Зададимся вопросом, какие из потенциальных реализаций модели обеспечат безусловное или условное выполнение действий a_{12} , a_{13} , a_{14} , a_{15} и a_{16} или, напротив, их невыполнение? Далее такую проверку будем называть проверкой на табуированность, которая сводится к установлению факта запрета или разрешения выполнения того или иного действия. Ответы на поставленные вопросы могут дать последовательности выполнения действий (рис. 2). Но те же ответы могут быть получены на основании анализа подграфов, в которые включаются все узлы и связи, влияющие на табуированность сворачивающего соединения (рис. 2).

Так, действие a_{12} (рис. 2, а) активизируется не при всех реализациях модели. Виной всему узел τ_2 , задающий альтернативу (в виде a_3 и a_4), которая исключает переход к a_7 , если выбрано a_4 . В этом, в частности, проявляется условность в выполнении процесса.

Построение и использование последовательностей действий может оказаться трудоемким занятием для графов большой размерности. Но если искусственно задать тип входа в виде логических \wedge или \vee для всех действий, включенных в модель, то провести соответствующий анализ будет намного проще. Такие входы, говорящие об альтернативности (вход \vee) или безальтернативности (вход \wedge) выполнения действий, подлежащих свертке, показаны на рис. 2 в виде символов \vee и \wedge , заключенных в кружки из штрихов.

Вход будет считаться альтернативным, если какой-либо из путей, ведущих к сворачиваемому действию, содержит хотя бы один разворачивающий узел со значением \vee или \vee . Все остальные входы будем полагать безальтернативными (вход \wedge). (Более подробно о способе определения того или иного варианта входа будет сказано позже. Пока же заметим, что использование еще одного варианта входа — входа \vee — бессмысленно, так как по

условию задачи требуется отразить факт выбора одного из двух взаимоисключающих событий: совершения или несвершения действия.)

В отличие от a_{12} действие a_{13} (рис. 2, б) никогда не будет выполнено, потому что для его инициализации необходимо сначала свернуть a_8 и a_9 , развернутые \vee -соединением, исключаящим, по определению, параллельное выполнение исходящих из \vee действий. То же самое можно сказать в отношении двух других действий a_{14} и a_{15} , но причина будет иная: в модели процесса отсутствует информация о том, по какому из двух путей должен дальше развиваться процесс после того, как завершилось a_{10} . Обе конструкции являются ошибочными и должны быть исключены из модели перед тем, как начнется ее проверка на табуированность.

Переходя к анализу подграфа, включающего a_{16} (рис. 2, в), следует сказать, что он корректен, но, как и в случае с a_{12} , все его действия выполняются на альтернативной основе.

Из показанного видно, что не каждое действие, вовлеченное в процесс, может произойти при всех потенциальных реализациях процесса.

Обобщая все, касающееся вариантов развития процесса, а именно его невыполнения, а также условной и безусловной реализации, будем далее различать синтаксически и семантически корректные процессы. Синтаксически корректными будут считаться те процессы, в которых отсутствуют запрещенные связи между узлами, в том числе такие, которые делают невозможным в принципе выполнение отдельных действий. Если говорить формально, то в сети отсутствуют пути, которые ведут к таким тупиковым, по сути, действиям.

В свою очередь, семантически корректными являются те процессы, которые исключают появление последовательностей, не соответствующих фактам, при всех реализациях этого процесса. Отсюда синтаксическую корректность можно интерпретировать как необходимое, а семантическую корректность как достаточное условия корректности сети.

В настоящей работе мы решаем две задачи. Сначала исследуем "естественные" ограничения, опреде-

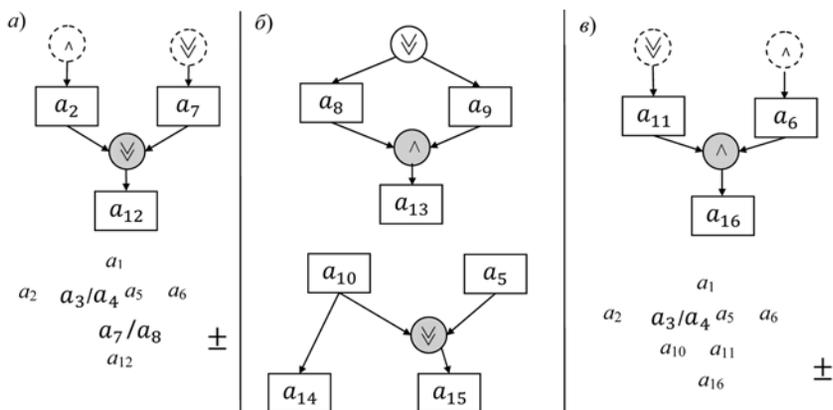


Рис. 2. Последовательности выполнения действий и структуры для анализа на табуированность сворачивающих соединений

ляющие синтаксическую корректность сети. Далее, рассматриваем классы сворачивающих соединений и разрабатываем метод проверки их на табуированность, обеспечивающий семантическую корректность процесса.

"Функциональные" ограничения

Как уже было отмечено, ориентированный граф — наиболее подходящая модель для описания структуры процесса. Все ограничения, образующие группу "функциональных" ограничений и множество "логических" ограничений, которые будут рассмотрены позже, следствие того, что узлы графа не могут быть связаны друг с другом произвольным образом. Например, одно и то же действие не может быть развернуто несколькими разворачивающими соединениями. В свою очередь, разворачивающее соединение не коммутируется со сворачивающим соединением и т. д. (Заметим, что все перечисленное — это аксиомы, которые, по определению, не доказываются, а могут быть сформулированы на основании постоянства фактов, полученных при исследовании организации бизнес-процессов). "Функциональные" ограничения можно описать посредством простого перечисления допустимых функциональных зависимостей между элементами множеств, образующих узлы процессного графа.

Введем следующие обозначения. Пусть $A = \{a_k\}_1^a$ — множество узлов-действий графа; $\bar{C} = \{\bar{c}_i\}_1^{\bar{c}}$ — множество разворачивающих соединений, присутствующих в графе; $\underline{C} = \{c_j\}_1^c$ — множество сворачивающих соединений. (Здесь и далее нижний индекс в обозначении множества ссылается на его первый элемент, а верхний индекс — на последний элемент, что является сокращенной формой записи выражения, задающего множество посредством перечисления его элементов.)

Ограничение 1. Каждое отдельно взятое действие может быть развернуто одним и только одним разворачивающим соединением. Это означает, что присутствует функциональная зависимость вида $f_1: A \rightarrow \bar{C}$. Поскольку в общем случае не все действия принимают участие в данном отображении, то f_1 — это частичная (нетотальная) функция. В то же время это сюръективная, но не инъективная функция.

Ограничение 2. Никакое действие не может быть свернуто более чем одним сворачивающим соединением. Отсюда следует, что должна иметь место функциональная зависимость $f_2: A \rightarrow \underline{C}$. Зависимость f_2 — это также сюръективная, но не инъективная функция. Кроме того, каждому элементу множества \underline{C} должно быть обязательно поставлено в соответствие более одного элемента множества A .

Ограничение 3. Отношения могут устанавливаться между элементами, образующими множество \bar{C} ,

что говорит о потенциальном существовании функциональной зависимости $f_3: \bar{C} \rightarrow \bar{C}$.

Ограничение 4. Между логическими узлами графа могут также присутствовать функциональные отображения вида $f_4: \underline{C} \rightarrow \bar{C}$.

Ограничение 5. Допустимы отношения между элементами множества A , что можно записать как $f_5 = A \rightarrow A$. f_5 — это, во-первых, в общем виде, частичная функция, поскольку отношения возникают далеко не между всеми действиями, образующими множество A , что подтверждается присутствием функциональных зависимостей f_1 и f_2 . И, во-вторых, f_5 — это биекция, так как все разветвления в WF-сетях моделируются только посредством использования логических узлов.

Формальное описание процессного графа будет неполным без указания двух функций: $f_{\bar{C}L}: \bar{C} \rightarrow L$ и $f_{\underline{C}L}: \underline{C} \rightarrow L$, где $L = \{\wedge, \vee, \forall\}$ — множество логических операций, с помощью которых можно найти тип (значение) логического узла.

Представленных ограничений недостаточно, чтобы исключить появление некорректных (с позиции здравого смысла) конструкций, например таких, какие показаны на рис. 3.

Чтобы различать два и более элемента, принадлежащих одному и тому же множеству, использована следующая форма записи. Если, например, i — индекс некоторого элемента множества, то $i \pm i'$, как и $i \pm i''$, — индексы других элементов, несопадающих друг с другом и с i -м элементом. Здесь i' , i'' и i — переменные, принимающие значения, принадлежащее множеству натуральных чисел \mathbb{N} . Очевидно, что при такой форме записи требуется задание ограничений в виде неравенств, показывающих, что значение индекса не превышает мощность соответствующего множества.

Недопустимы (абсурдны) отношения между парами разворачивающих узлов, принимающих одинаковые значения (рис. 3, а). Поскольку $f_3 = \{(\bar{c}_i, \bar{c}_{i \pm i'}) \in \bar{c}^2 \mid \bar{c}_i = f_3(\bar{c}_{i \pm i'}) \wedge \bar{c}_i \neq \bar{c}_{i \pm i'}\}$, то очередное ограничение можно формально представить следующим образом.

Ограничение 6. Для $\forall(\bar{c}_i, \bar{c}_{i \pm i'})$ должно выполняться $f_{\bar{C}L}(\bar{c}_i) \neq f_{\bar{C}L}(\bar{c}_{i \pm i'})$.

Другая конструкция, приведенная на рис. 3, б, также бессмысленна. Для ее исключения сформу-

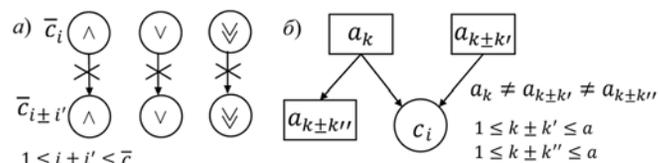


Рис. 3. Некоторые запрещенные подграфы сценарных моделей бизнес-процессов

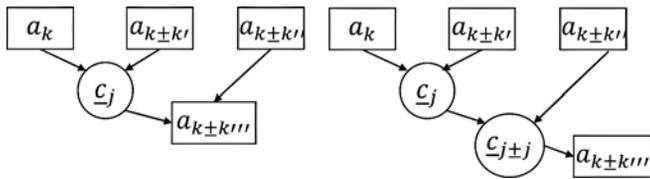


Рис. 4. Семантически корректные варианты сворачивающих соединений

лируем ограничение, манипулирующее областью определения ранее введенных функций.

Ограничение 7. $Domf_2 \cap Domf_3 = \emptyset$. Никакое действие не может одновременно участвовать в свертке и выступать в роли предка (предшественника) какого-либо другого действия.

Обратим внимание на то, что рассмотренные здесь ограничения не исключают присутствие каких-то иных классов функциональных зависимостей между элементами трех множеств A , \bar{C} и C , образующих узлы процессного графа. На рис. 4 приведены примеры двух подграфов с "участием" семантически корректных сворачивающих соединений, которые не формализуются посредством перечисленных ограничений.

Далее ограничимся анализом только подграфов, соответствующих заданным ограничениям, полагая, что продемонстрированные на рис. 4 варианты соединений могут быть приведены к виду, удовлетворяющему введенным функциональным зависимостям.

Классы и подклассы соединений

Как было установлено ранее, проверке на табуированность (запрет или разрешение) должны подвергаться только узлы, образующие множество C . Логично все комбинации соединений разбить на три класса (типа) по числу используемых вариантов сворачиваемых логических операций: конъюнкции, дизъюнкции и строгой дизъюнкции. Определимся с вариантами структур, содержащих сворачивающий узел и "прикрепленные" к нему (входящие в него) действия. Такая структура должна включать все элементы, оказывающие влияние на результаты свертки. К этим элементам, что было выявлено ранее, следует отнести также классы соединений, посредством которых действия были развернуты, а также виды входов для всех разворачивающих соединений: логические \wedge или \vee .

Самая простая (элементарная) конфигурация — это, когда в свертке участвуют действия, принадлежащие одному и тому же разворачивающему соединению (рис. 5, а). При этом неважно, какая из разновидностей входов используется для разворачиваемого соединения: вход \wedge или вход \vee , поскольку на результаты последующей свертки это повлиять никак не может.

Сложные типы конфигурации возникают, когда свертке подвергаются действия, развернутые раз-

ными соединениями. Отдельно следует выделить два сложных подтипа. Первый характеризуется тем, что в свертке участвуют по одному действию от каждого развернутого соединения (рис. 5, б). Второй подтип отличается тем, что может включать также и действия, которые были развернуты одним и тем же соединением (рис. 5, в). Ниже будет показано, что каждый из трех конфигурационных вариантов по-разному влияет на конечный результат — полную или частичную табуированность сворачивающего соединения.

О влиянии входов разворачивающих соединений (логических \vee или \wedge) на результаты свертки действий говорилось в предыдущем разделе. Сформулируем правило, следуя которому можно определить тип (вариант) входа (вход \vee или вход \wedge) для разворачивающего узла.

Правило. Если какой-либо из путей, соединяющих исходную вершину процессного графа с некоторым разворачивающим узлом, для которого определяется вариант входа, содержит хотя бы один разворачивающий узел, для которого $f_{CL}(C) = \vee$ или $f_{CL}(C) = \vee$, то входом будет логическое \vee . В противном случае вход интерпретируется как \wedge .

Нетрудно показать, что при решении таких задач как, например, нахождение множества альтернативных путей в графе или определение достижимости того или иного узла-действия, фактор входа актуален только для комбинации "вход \vee — разворачивающий узел \wedge ", так как для всех остальных допустимых сочетаний значений входов и разворачивающих узлов присутствие альтернативы в выполнении или невыполнении действия очевидно, поскольку разворачивающие соединения \wedge или \vee "альтернативны" по определению.

Выделим отдельные подклассы внутри каждого из трех классов соединений и перечислим виды таких подклассов. Будем ориентироваться на то, что подкласс характеризуется уникальным сочетанием значений, принимаемых разворачивающими узлами и их входами. Принимая во внимание сказанное выше, получим следующие варианты подклассов для каждого из классов:

- \vee, \vee, \wedge -подклассы; все разворачивающие узлы в подклассе имеют одинаковые \vee, \vee или \wedge значения;

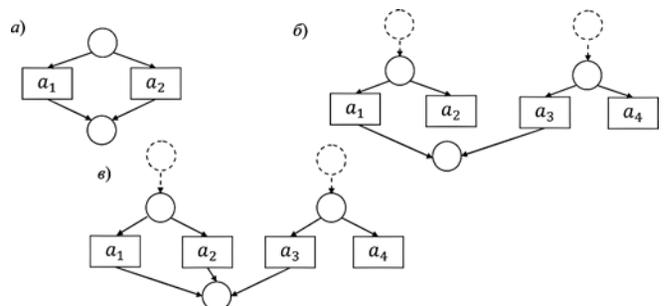


Рис. 5. Подграфы типовых конфигураций

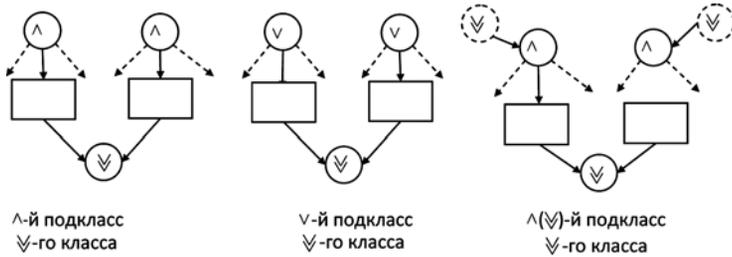


Рис. 6. Примеры подклассов \forall -го класса

- $\wedge\forall, \forall\forall, \wedge\forall, \forall\wedge\forall$ -подклассы; разворачивающие узлы принимают соответственно комбинации значений \wedge и \vee, \forall и \vee, \wedge и \forall, \forall и \wedge и \forall ;
- $\wedge(\forall), \wedge(\forall)\vee, \wedge(\forall)\forall, \wedge(\forall)\forall\forall$ -подклассы, в которых входами для \wedge -соединений выступают \forall -входы.

Примеры таких подклассов для \forall -класса приведены на рис. 6.

Далее на примере некоторых подклассов \forall -класса продемонстрируем метод анализа сворачивающих соединений на табуированность. Использование данного метода позволяет алгоритмизировать этот процесс и не рассматривать по отдельности все конфигурации свертки для каждого из подклассов соединений.

Некоторые допустимые и табуированные соединения \forall -класса

Как было уже установлено, вследствие альтернативности, обусловленной наличием разворачивающих и сворачивающих соединений, часть действий может как произойти, так и не произойти, а какие-то, ввиду присутствия синтаксической некорректности в структуре процесса, могут вообще никогда не состояться. Сказанное можно представить и формально, воспользовавшись такими булевыми функциями, как конъюнкция, дизъюнкция и сложение по модулю 2, которые "соответствуют" логическим условиям AND, OR и XOR. Это функции n -переменных, число которых равно числу действий, участвующих в свертке. Запись $a_k = 1$ будет означать, что действие произойдет (произошло), $a_k = 0$ — прямо противоположное. Очевидно, что если соответствующая булева функция принимает значение 1, то сворачивающее соединение "открыто" и могут быть выполнены все действия и (или) сработают все логические условия, следующие за ним. В противном случае, при значении 0, процесс должен завершиться в этой точке.

Результаты анализа на табуированность в виде таблиц истинности полного набора типовых конфигураций \forall -класса с разворачивающими \wedge -узлами показаны на рис. 7. Для комбинации "развора-

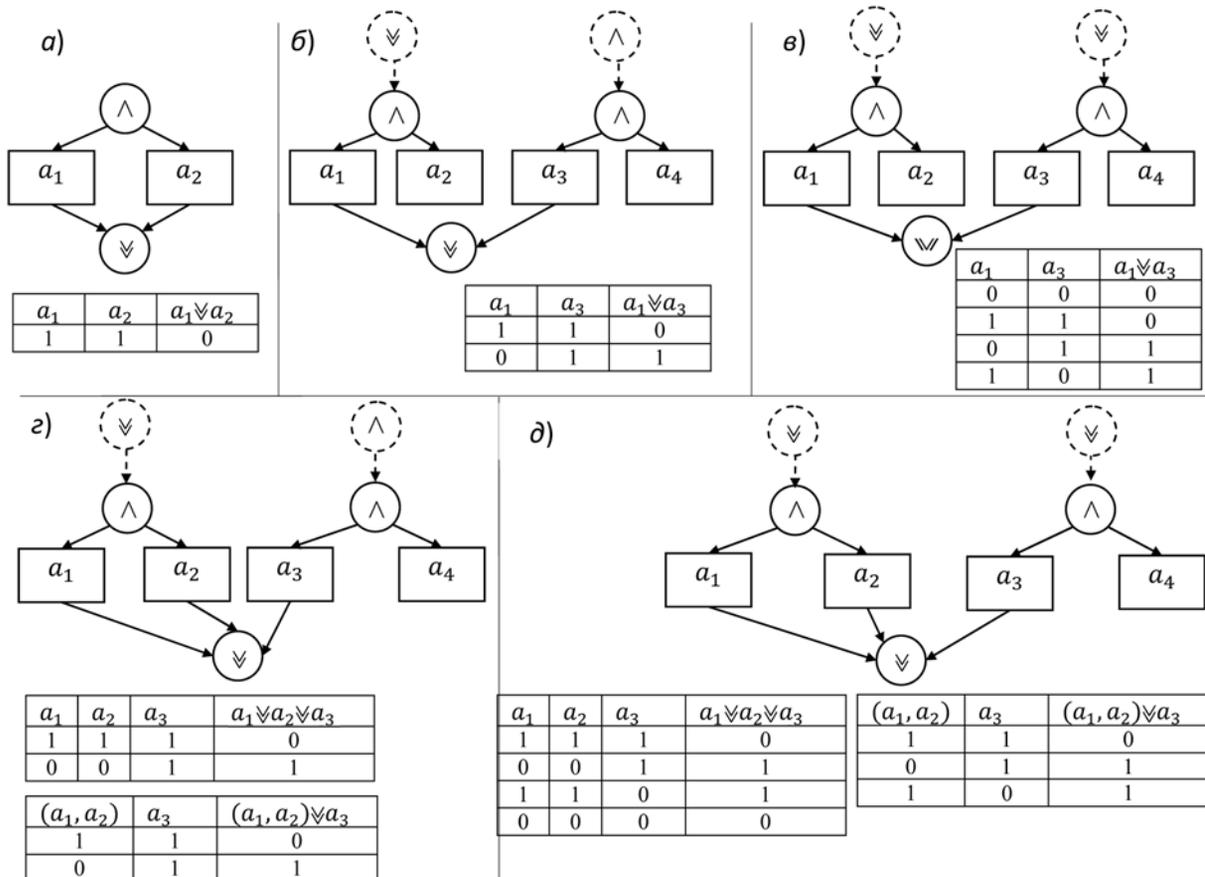


Рис. 7. Анализ на табуированность соединений \forall -класса

сворачивающее \wedge — сворачивающее \vee " соединением единственной запрещенной конструкцией будет та из них, что показана на рис. 7, а. Действительно, два и более действия не могут быть свернуты логическим \vee на основании обязательности их совместного выполнения, вытекающего из того, что все они были инициированы ранее разворачивающим \wedge . Поэтому таблица "истинности" для конструкции, показанной на рис. 7, а, включает всего одну единственную строку с $a_1 = 1$ и $a_2 = 1$.

Несколько иная картина будет наблюдаться, если операнды сворачивающего соединения принадлежат различным разворачивающим \wedge . В этом случае вид таблицы истинности будет зависеть также и от вида входа разворачивающего узла.

Пусть одному разворачивающему соединению предшествует \wedge (выступает в качестве входа в конструкцию), а другому — \vee (рис. 7, б). В этом случае в соответствии с получаемой таблицей истинности сворачивающее соединение может присутствовать в модели, но только половина ее реализаций будут возможны (когда $a_1 = 0$ и $a_2 = 1$).

Аналогичная ситуация возникает, когда число операндов, участвующих в соединении, увеличивается, например до трех, но два из них относятся к одному и тому же разворачивающему \wedge (рис. 7, з). В этом случае, если два и более действия принадлежат одному и тому же разворачивающему соединению, то они могут быть заменены одним. Получаемые таблицы истинности эквивалентны. Число строк в последних увеличится (станет максимальным), если обоим разворачивающим соединениям \wedge будут предшествовать \vee (рис. 7, в, д).

Проведение анализа на табуированность всех классов и подклассов соединений — трудоемкая и рутинная процедура, которую можно алгоритмизировать. Для этого должны быть известны вид булевой функций (что не составляет труда определить на основании значения сворачиваемого соединения), число булевых переменных и область их определения.

Для нахождения двух последних компонентов проведем очевидное преобразование анализируемых подграфов (см. рис. 7), приведя их к виду, показанному на рис. 8. Для наглядности все изменения выделены серым цветом.

В процессе трансформирования мы руководствовались следующими правилами и последовательностью их выполнения.

1. Если в свертке участвуют действия, развернутые каким-либо логическим \wedge , то все они должны интерпретироваться как одно действие.

2. Значения всех разворачивающих \wedge -узлов, имеющих в качестве входов логическое \vee , должны быть заменены на \vee .

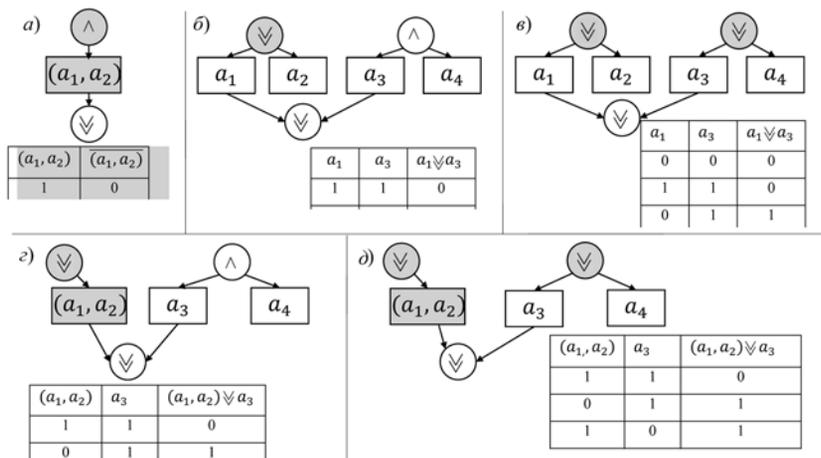


Рис. 8. Подграфы, полученные в результате преобразования исходных конструкций

3. В соединениях \vee -класса при свертке действий, принадлежащих одному и тому же разворачивающему \wedge -узлу, следует использовать булеву функцию отрицание (см. рис. 7, а).

Как нетрудно заметить, таблицы истинности для исходных (см. рис. б) и преобразованных (см. рис. 7) подграфов идентичны.

Далее, для нахождения числа переменных булевых функций воспользуемся обратным отображением F_1^- для функции $a = f_1(\underline{c})$. Поскольку \underline{c} всегда известно, не представляет сложности найти искомого число переменных.

С учетом того, что переменные, соответствующие действиям, развернутым \wedge -соединениями, могут принимать единственное значение, которое всегда равно 1, число строк в таблице истинности составит $2^n - n'$, где n' — число \wedge -соединений в анализируемой конструкции.

Непосредственно сам анализ на табуированность после последовательного выполнения всех перечисленных пунктов сводится, по сути, к интерпретации полученной совокупности значений соответствующей булевой функции. Возможны три варианта исхода:

— все элементы множества значений функции равны 0; это означает, что данная конструкция ошибочна и не может присутствовать в модели;

— все элементы принимают значение 1, что свидетельствует о безусловной реализуемости последующих действий, непосредственно ассоциированных со сворачиваемым узлом;

— часть элементов равна 1, а часть 0; следовательно, имеет место случай условной реализуемости процесса, представленной WF-сетью, когда некоторые пути заканчиваются тупиками.

Обратим внимание на то, что приемлемость или отрицание того или иного исхода может быть установлена только на основании целевой семантики конкретного процесса.

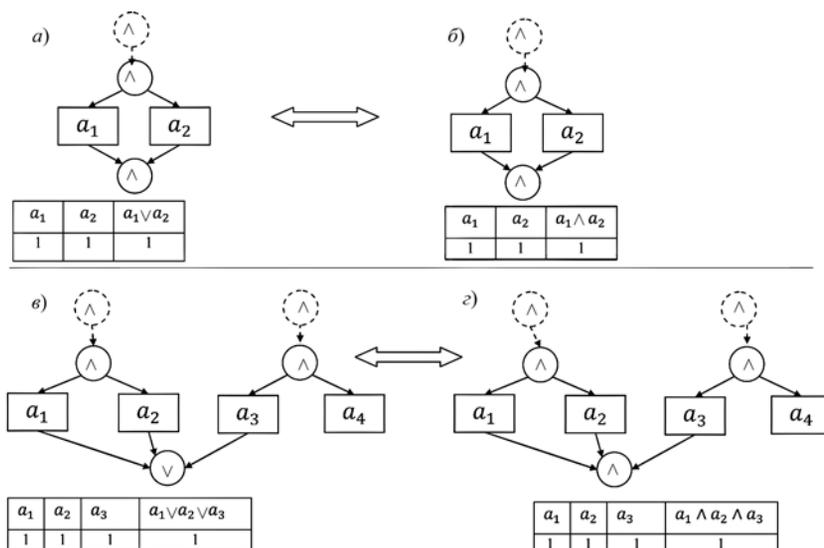


Рис. 9. Примеры эквивалентных соединений

Эквивалентные конструкции

В сценарных моделях можно обнаружить тождественные подклассы соединений, один из которых будет всегда относиться к \wedge -классу. Конструкции, показанные на рис. 9, а и в, — абсурдны, если подходить к ним с позиции здравого смысла. Действительно, два и более действий, актуализированных \wedge -соединением, не имеет никакого смысла сворачивать логическим OR. Приемлемое решение — задействовать сворачивающие \wedge -коннекторы (рис. 9, б и з).

Использование подобного приема в незначительной степени, но снижает остроту проблемы неоднозначности, связанной с использованием OR-соединений, когда при достижении сворачивающего OR-узла одним из действий, параллельно выполняемых ему последовательности действий еще не завершились, и требуется принять решение о целесообразности их последующего выполнения.

Заключение

Разработан метод быстрой проверки сценарных моделей бизнес-процессов на бездефектность, который может быть использован как во время их построения, так и последующей реорганизации. Для верифицируемых WF-сетей должны выполняться все функциональные ограничения, которые исключают появление неинтерпретируемых (в семантическом контексте) связей между узлами. Вводимые ограничения описывают хорошо изученную семантику наиболее часто применяющихся в практике проектирования управляющих конструкций, но, в то же время, оставляют открытыми все вопросы, связанные с использованием OR-соединений. Тем не менее, опираясь на предлагаемый порядок

проверки корректности сети, можно обнаружить явные и условные (относительные) тупики для всех потенциальных ее конфигураций.

Полученные результаты — основа дальнейшего совершенствования организации процессно-ориентированных информационных систем. Если расширить границы корректности сети и бездефектными считать процессы, в которых не только отсутствуют замкнутые циклы, тупики и нетерминальные связи, но и которые гарантируют получение заданных последовательностей фактов, то процедуру верификации следует дополнить функциями проверки достижимости и (или) недостижимости таких последовательностей. В последнем случае сеть потоков работ может оказаться просто бесполезной.

При подобной постановке вопроса появляется возможность решать некоторые задачи достоверности информации, сосредоточенной в электронных базах данных. В частности, при обнаружении несоответствия или ошибки в уже зафиксированном факте, можно локализовать всю цепочку фактов, связанных с этим фактом, и исправить те из них, которые оказались ошибочными.

Список литературы

1. Santos P. S., Almeida J., Guizzardi G. An ontology — based analysis and semantics for organizational structure modeling in the ARIS method // Data and knowledge engineering. 2013. Vol. 38. P. 690—708.
2. Родионов А. Н. Системная интерпретация ключевых категорий концептуальных моделей баз данных // Информационные технологии. 2016. № 6. Том 22, № 6. С. 447—557.
3. Ван дер Аалст В., ван Хей К. Управление потоками работ: модели, методы и системы. М.: ФИЗМАТЛИТ, 2007. 316 с.
4. Van der Aalst W. M. P. Formalization and verification of event-driven process chains // Information and software technology. 1999. Vol. 41, N. 10. P. 639—650.
5. Van der Aalst W. M. P., van Hee K. M., ter Hofstede A. H. M., Sidorova N., Verbeek H. M. W., Voorhoeve M., Wynn M. T. Soundness of workflow nets: classification, decidability, and analysis // Formal aspects of computing. 2011. Vol. 23 (3). P. 333—363.
6. Cravo G. Applications of propositional logic to workflow analysis // Applied mathematics letters. 2010. Vol. 23. P. 272—276.
7. Verbeek H. M. W., van der Aalst W. M. P., Wil M., ter Hofstede, Arthur H. M. Verifying workflows with cancellation regions and OR-joins: An approach based on relaxed soundness and invariants // The computer journal. 2007. Vol. 50, N. 3. P. 294—314.
8. Kindler E. On the semantics of EPCs: A framework for resolving the vicious circle // Data and knowledge engineering. 2006. Vol. 56 (1). P. 23—40.
9. Wynn M. T., van der Aalst W. M. P., ter Hofstede A. H. M., Edmond D. Verifying workflows with cancellation regions and OR-joins: An approach based on reset nets and reachability analysis // International Conference on Business Process Management (BPM 2006), vol. 4102 of Lecture Notes in Computer Science. P. 389—394. Berlin: Springer-Verlag, 2006.

Some Syntactic and Semantic Limits on Scenary Business Process Models

The main goal of this paper is to obtain the simple and descriptive method for base workflows verification on soundness. This method can be used during immediate development or reengineering of workflow models. Formally, any workflow can be described as graph that contains vertices of two types: actions and logical connectors. By means of the latter is given alternativeness of instantiation at the business process. We expand the standard conditions of soundness such as absence of dangling references, deadlocks and livelocks and consider that workflow model is sound if all initial facts for sake of model is a progress will be derived after process termination.

In order to eliminate unreal graph links we impose some functional restrictions on the potential maps between elements of the finite sets such as the set of actions, the set of join connectors and the set of split connectors. The syntactic control comes down to the checking existence of only given restrictions. The semantic control assumes that a workflow net doesn't contain livelocks. We propose the method using Boolean variants and functions for rapid unveiling total and contingent livelocks at workflow models. Join connectors are the points of expectative livelocks. Under the total livelocks any paths that pass through the join nodes are blocked and the business process is terminated. In case of contingent locking only some of the paths aren't interrupted.

As a result, syntactic and semantic controls are correspondently necessary and sufficient conditions of net correctness.

Keywords: workflow net, soundness, verification, livelocks, syntactic and semantic correctness, syntactic and semantic control

References

1. Santos P. S., Almeida J., Guizzardi G. An ontology —based analysis and semantics for organizational structure modeling in the ARIS method, *Data and knowledge engineering*, 2013, vol. 38, pp. 690—708.
2. Rodionov A. N. Sistemnaja interpretacija ključevyh kategorij konceptu-al'nyh modelej baz dannyh, *Informacionnye tehnologii*, 2016, vol. 22, no. 6, pp. 447—557. (Rodionov A. N. System interpretation of conceptual data model key categories, *Information technologies*, 2016, vol. 22, no. 6, pp. 447—557) (in Russian).
3. Vil van der Aalst W., vann Hej K. *Upravlenie potokami rabot: modeli, metody i sistemy*, Moscow, FIZMATLIT, 2007, 316 p. (Wil van der Aalst, Kees van Hee Workflow management: models, methods, and systems. Moscow: FIZMATLIT, 2007, 316 p.) (in Russian)
4. Van der Aalst W. M. P. Formalization and verification of event-driven process chains, *Information and software technology*, 1999, vol. 41 (10), pp. 639—650.
5. Van der Aalst W. M. P., van Hee K. M., ter Hofstede A. H. M., Sidorova N., Verbeek H. M. W., Voorhoeve M., Wynn M. T. Soundness of workflow nets: classification, decidability, and analysis, *Formal aspects of computing*, 2011, vol. 23, no. 3, pp. 333—363.
6. Cravo G. Applications of propositional logic to workflow analysis, *Applied mathematics letters*, 2010, vol. 23, pp. 272—276.
7. Verbeek H. M. W., van der Aalst W. M. P., Wil M., ter Hofstede, Arthur H. M. Verifying workflows with cancellation regions and OR-joins: An approach based on relaxed soundness and invariants, *The computer journal*, 2007, vol. 50 (3), pp. 294—314.
8. Kindler E. On the semantics of EPCs: A framework for resolving the vicious circle, *Data and knowledge engineering*, 2006, vol. 56 (1), pp. 23—40.
9. Wynn M. T., van der Aalst W. M. P., ter Hofstede A. H. M., Edmond D. Verifying workflows with cancellation regions and OR-joins: An approach based on reset nets and reachability analysis, *International Conference on Business Process Management (BPM 2006)*, vol. 4102 of Lecture Notes in Computer Science. Berlin, Springer-Verlag, 2006, pp. 389—394.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.056.53

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

Подход к моделированию параметров и характеристик интегрированной информационно-эксплуатационной безопасности

Исследованы вопросы математического моделирования современных защищенных отказоустойчивых информационных систем в комплексе. Сформулирована задача моделирования, определен математический аппарат, используемый для моделирования, предложены основные подходы к моделированию параметров и характеристик интегрированной информационно-эксплуатационной безопасности информационных систем — параметры и характеристики надежности и безопасности информационных систем. Данные параметры и характеристики определяются на одной комплексной модели, позволяющей учитывать их взаимное влияние. Предложенные подходы к моделированию могут применяться при построении отказоустойчивых защищенных информационных систем.

Ключевые слова: безопасность информационной системы, безопасность информации, угроза атаки, угроза уязвимостей, отказ элемента информационной системы, надежность, резервирование, математическое моделирование, комплексная оценка, марковская модель, интегрированная безопасность

Введение

При проектировании современных информационных систем повышение уровней их надежности и защищенности рассматривается как сопоставимо важные задачи. В обоих случаях присутствует понятие отказа. С точки зрения защиты информации — это возникновение в системе реальной угрозы атаки, которой может воспользоваться потенциальный нарушитель, также присутствует понятие резервирования — резервирование угроз уязвимостей информационной системы угрозами уязвимостей средства защиты информации и т.д. [1, 2]. Решение этих задач по отдельности не эффективно, поскольку результаты решения задачи повышения уровня надежности могут входить в противоречие с результатами решения задачи повышения уровня защищенности. Это достаточно наглядно проиллюстрировано в работе [3], где исследован вопрос структурного резервирования элементов информационной системы и его влияния на уровень защищенности информационной системы. В результате данного исследования, где с позиций информационной безопасности структурное резервирование интерпретируется как резервирование элементов по угрозам атак, показано, что применение известных из теории надежности методов структурного резервирования в информационной системе приводит к снижению уровня ее защищенности, что

потребовало разработки принципиально иного подхода к структурному резервированию элементов [4]. С учетом проведенных исследований был сделан вывод о необходимости решения данных задач в комплексе в целях построения отказоустойчивых защищенных информационных систем.

В работе [3] было введено понятие интегрированной информационно-эксплуатационной безопасности, определенной как свойство информационной системы сохранять во времени в установленных пределах значения всех характеристик безопасности информации, определяющих способность выполнять информационной системой требуемые функции обработки информации в безопасном режиме, в условиях отказов элементов информационной системы и реализации атак на информационную систему потенциальным нарушителем в процессе ее эксплуатации.

В рамках решения данных задач в комплексе при проектировании отказоустойчивых защищенных информационных систем необходимо построение комплексных моделей, позволяющих определять не параметры и характеристики надежности и безопасности по отдельности, без последующего понимания того, как одни влияют на другие, а именно параметров и характеристик интегрированной информационно-эксплуатационной безопасности.

Подходы к моделированию параметров и характеристик безопасности

В работе [1] изложен подход к моделированию угрозы атаки марковской моделью с дискретными состояниями и непрерывным временем с последующим построением для нее модели вероятностного разрежения входных потоков случайных событий. Подход основан на введенной в работе [5] интерпретации угрозы атаки схемой параллельного резервирования создающих угрозы атаки угрозами уязвимостей. Корректность использования марковских процессов при моделировании обоснована в [6]. В результате реализации данного подхода к моделированию могут быть рассчитаны надежные параметры и характеристики безопасности угрозы атаки, поскольку при этом моделируется не атака, как процесс последовательного деструктивного воздействия нарушителем на систему, а именно угроза атаки, как процесс возникновения и устранения в системе отказов информационной безопасности — реальных угроз атак, что позволило ввести в [1] понятие надежности информационной безопасности. Также в [1] сформулированы требования к корректности построения для решаемых задач моделирования марковских моделей с дискретными состояниями и непрерывным временем, предложен подход к объединению состояний в модели вероятностного разрежения входных потоков случайных событий, существенно расширяющий возможности моделирования в части определения различных надежных характеристик безопасности угроз атак.

Проиллюстрируем данный подход к моделированию примером.

Исходными данными при моделировании выступают интенсивность возникновения уязвимости λ и интенсивность устранения уязвимости μ , которые могут быть объективно, без использования каких-либо экспертных оценок, определены из соответствующей статистики [1]. Задачей же моделирования является определение параметров и характеристики безопасности угрозы атаки, создаваемой соответствующей совокупностью (набором) угроз уязвимостей реализации, таких как интенсивность возникновения и устранения в системе реальной угрозы атаки λ_a и μ_a , вероятности готовности к безопасной эксплуатации информационной системы в отношении угрозы атаки P_{0a} , среднего времени наработки на отказ безопасности информационной системы (восстанавливаемая система) в отношении угрозы атаки $T_{0y,a}$ среднего времени восстановления безопасности информационной системы $T_{в.у,a}$ в отношении угрозы атаки, которые могут рассматриваться в качестве количественных оценок актуальности угрозы атаки при последующем проектировании системы защиты.

Пусть угроза атаки создается двумя угрозами (типами угроз) уязвимостей, с соответствующими

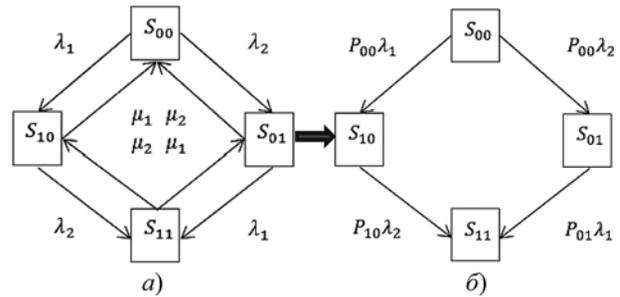


Рис. 1. Размеченные графы системы состояний случайного процесса для модели угрозы атаки: а — марковская модель; б — модель вероятностного разрежения входных потоков

параметрами — интенсивностями выявления и устранения уязвимостей (аналогичным образом можно построить модель для угрозы атаки любой сложности). Состояния системы обозначим S_{ij} , где i — число выявленных уязвимостей первого типа, j — число уязвимостей второго типа. Размеченный граф системы состояний случайного процесса (марковского процесса) приведен на рис. 1, а.

Построение модели вероятностного разрежения входных потоков основано на следующих соображениях [1]. Входной поток событий, поступающий в систему (на вход марковской модели), вероятно разрежается — распределяется между состояниями системы, в том смысле, что событие может наступить в случайный момент времени, в который система находится в одном из возможных своих состояний (переходы между состояниями в марковской модели осуществляются мгновенно).

С использованием модели вероятностного разрежения входных потоков параметр безопасности угрозы атаки λ_a может быть рассчитан по следующей формуле:

$$\lambda_a = \sum_{S_i \in S_{(R-1)}} P_{S_i} \lambda_{S_i, S_R},$$

где $S_{(R-1)}$ — множество состояний системы, характеризующих отсутствием в системе реальной угрозы атаки (не все создающие угрозу атаки уязвимости выявлены и устранены), в каждом из которых система находится с вероятностью $P_{S_{(R-1)}}$; S_R — состояние возникновения в системе реальной угрозы атаки (все создающие угрозу атаки уязвимости выявлены и не устранены). В состоянии S_R из состояний $S_{(R-1)}$ в системе осуществляется переход с интенсивностью $\lambda_{S_{(R-1)}, S_R}$.

Поскольку в стационарном (установившемся) режиме функционирования системы за долю времени нахождения системы в состоянии, характеризуемом возникновением реальной угрозы атаки, определяемым, как $1 - P_{0a}$, где P_{0a} — вероятность готовности системы к безопасной эксплуатации в отношении угрозы атаки, из состояния, характери-

зующего реальную угрозу атаки, исходит поступающий в него поток событий интенсивностью λ_a (система без потерь, все выявляемые уязвимости устраняются). Сказанное позволяет следующим образом рассчитывать параметр безопасности μ_a угрозы атаки:

$$\mu_a = \frac{\lambda_a}{1 - P_{0a}}.$$

Соответствующим образом можно определить и надежностные характеристики безопасности угрозы атаки.

Вероятность готовности информационной системы к безопасной эксплуатации в отношении угрозы атаки для модели (см. рис. 1)

$$P_{0a} = P_{00} + P_{10} + P_{01},$$

среднее время наработки на отказ безопасности информационной системы (восстанавливаемая система) в отношении угрозы атаки $T_{0y.a}$

$$T_{0y.a} = \frac{1}{\lambda_a} - T_{в.у.а},$$

среднее время восстановления безопасности информационной системы $T_{в.у.а}$ в отношении угрозы атаки:

$$T_{в.у.а} = \frac{1}{\mu_a}.$$

Существенно расширяются возможности моделирования за счет объединения состояний в модели вероятностного разрежения входных потоков случайных событий [1]. Проиллюстрируем сказанное примером. Пусть построена марковская модель, представленная на рис. 2, а. Ее особенность состоит в том, что в системе присутствуют два состояния возникновения реальных угроз атак: S_{11} — в системе присутствует одна реальная угроза атаки и S_{21} — в системе присутствуют одновременно две реальные угрозы атаки. Построив модель вероятност-

ного разрежения входных случайных событий (по аналогии с тем, как это было проиллюстрировано на рис. 1), можем определить параметры безопасности — интенсивность возникновения в системе одной реальной угрозы атаки λ_{a1} и интенсивность возникновения в системе одновременно двух реальных угроз атак λ_{a2} :

$$\lambda_{a1} = P_{10}\lambda_2 + P_{01}\lambda_1;$$

$$\lambda_{a2} = P_{20}\lambda_2 + P_{11}\lambda_1.$$

Построим модель для оценки события — в системе присутствует хотя бы одна реальная угроза атаки, т. е. система находится в состоянии отказа безопасности, обозначим это состояние S_2 . Построение соответствующей модели состоит в объединении состояний S_{11} и S_{21} (при этом $P_2 = P_{11} + P_{21}$), в результате чего получаем модель вероятностного разрежения входных потоков случайных событий, приведенную на рис. 2, б.

Замечание. Требование к корректности объединения состояний в модели сформулированы в работе [1].

Обозначим параметры отказов (в системе присутствует, по крайней мере, одна реальная угроза атаки) и восстановлений (все реальные угрозы атак в системе устранены) безопасности информационной системы в отношении угрозы атаки как λ_0 и $\mu_в$. Для модели, приведенной на рис. 2, они могут быть рассчитаны следующим образом:

$$\lambda_0 = \lambda_2(P_{10} + P_{20}) + P_{01}\lambda_1;$$

$$\mu_в = \frac{(P_{10} + P_{20})\lambda_2 + P_{01}\lambda_1}{P_2} = \frac{(P_{10} + P_{20})\lambda_2 + P_{01}\lambda_1}{P_{11} + P_{21}},$$

где параметр $\mu_в$ рассчитывается с учетом того, что безопасность системы, нарушаемая с интенсивностью λ_0 , восстанавливается за долю времени $P_2 = P_{11} + P_{21}$ (это доля времени нахождения системы в объединенном состоянии S_2).

Соответствующим образом, как это было рассмотрено ранее, рассчитываются и требуемые характеристики безопасности информационной системы в отношении угрозы атаки.

В отличие от моделирования надежностных параметров и характеристик безопасности, при моделировании эксплуатационных параметров и характеристик безопасности [2], где уже моделируется процесс реализации потенциальным нарушителем возникающих в системе реальных угроз атак, строятся марковские модели с дискретными состояниями и непрерывным временем с потерями входных случайных событий. Вопросы корректности их построения исследованы в работе [2].

Иллюстрация построения подобной модели для рассмотренного ранее примера (см. рис. 1) представлена на рис. 3. Состояниями системы здесь уже характеризуется не возникновение реальной угро-

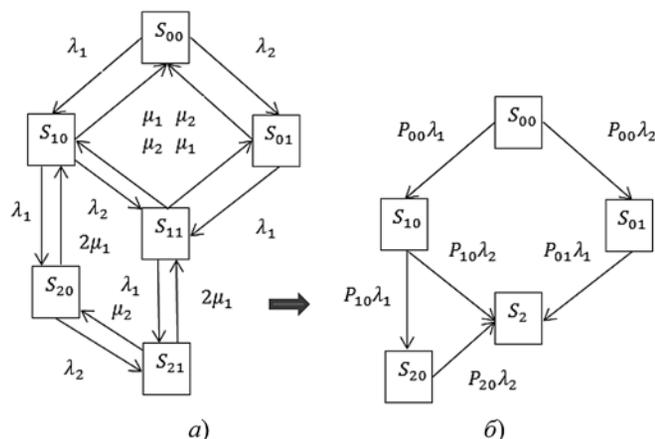


Рис. 2. Иллюстрация объединения состояний в модели вероятностного разрежения входных потоков случайных событий: а — исходная марковская модель угрозы атаки; б — модель вероятностного разрежения входных потоков с объединением состояний

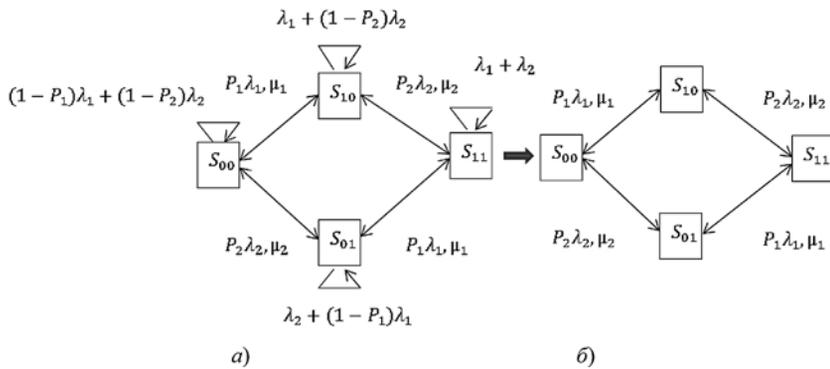


Рис. 3. Размеченные графы системы состояний случайного процесса для модели реализации угрозы атаки:
 а — включение в модель потоков потерь; б — марковская модель с потерями

зы уязвимостей, а реализация успешной атаки на возникшую в системе реальную угрозу уязвимостей.

Поток потерь (рис. 3, а) создают два потока случайных событий. Первый из них составляет события отказа безопасности, которым не может воспользоваться потенциальный нарушитель ввиду его неготовности к реализации атаки на возникшую в системе реальную угрозу уязвимостей, которая определяется вероятностью P_j . Подход к моделированию характеристики P_j , не требующий использования каких-либо экспертных оценок, изложен в работе [7]. Данные потери определяются условием $(1 - P_j)\lambda_j$ (рис. 3, а).

Второй поток потерь случайных событий определяется одновременным (не одномоментным) возникновением в системе двух и более угроз уязвимостей одного типа при условии, что первая возникшая из них реализована потенциальным нарушителем. Отнесение данных событий к событиям потерь (потерь для потенциального нарушителя) обуславливается тем, что ему незачем реализовывать несколько реальных угроз уязвимостей одного типа, предполагающих одно и то же последующее развитие атаки на информационную систему.

В рамках решения задачи моделирования реализации угроз атак [2] рассмотрены особенности моделирования безопасности информационной системы (восстанавливаемая система) и безопасности информации применительно к реализации угроз атак, направленных на нарушение конфиденциальности информации. Нарушение конфиденциальности информации уже представляет собой фатальный отказ (не восстанавливаемая система), поскольку конфиденциальность похищенной информации не может быть восстановлена. При моделировании в данном случае используются поглощающие состояния [2] — состояния реализации реальной угрозы атаки, из которых нет переходов. Для рис. 3 — это состояние S_{11} . В данном случае уже моделируется не среднее время наработки на отказ, а среднее время наработки до отказа без-

опасности информации (до нарушения ее конфиденциальности).

Характеристики безопасности (с соответствующей иной их интерпретацией — моделируется не угроза атаки, а процесс реализации угрозы атаки потенциальным нарушителем) определяются аналогично тому, как было описано ранее, на построенной для марковской модели с потерями модели вероятностного разрежения входных потоков случайных событий. При этом также при моделировании может реализовываться объединение состояний, что проиллюстрировано на рис. 2 [2].

Подход к моделированию параметров и характеристик интегрированной информационно-эксплуатационной безопасности

С точки зрения моделирования параметров и характеристик интегрированной информационно-эксплуатационной безопасности принципиально важным является то, что для моделирования параметров и характеристик безопасности и характеристик надежности для моделирования восстанавливаемых систем [8] используется один и тот же математический аппарат — марковские модели с дискретными состояниями и непрерывным временем. Отличия данных моделей состоит лишь в различной интерпретации состояний на графе переходов — отказ безопасности либо отказ надежности.

Например, модель, приведенная на рис. 1, а, используется для расчета характеристик надежности схемы резервирования.

Исходными данными при моделировании в этом случае выступают интенсивность возникновения отказа элемента λ и интенсивность восстановления элемента после отказа μ .

Состояния системы обозначены S_{ij} , где $i = 1$ обозначает отказ резервируемого элемента, $j = 1$ — отказ резервирующего элемента (резервирование реализуется по схеме "горячего резерва"). На рис. 1, а промоделирован случай, когда резервируемый и резервирующий элементы различны (для них различны интенсивности возникновения отказа элемента λ и интенсивности восстановления элемента после отказа μ).

В случае если резервируемый и резервирующий элементы полностью совпадают, модель, приведенная на рис. 1, а, преобразуется в модель, представленную на рис. 4.

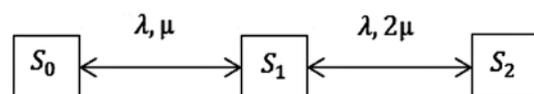


Рис. 4. Размеченный граф системы состояний случайного процесса для схемы резервирования одинаковыми элементами

Через S_j в данном случае обозначается число одновременно отказавших элементов.

Вместе с тем, с позиций интегрированной информационно-эксплуатационной безопасности эти состояния могут интерпретироваться аналогичным образом в том смысле, что они характеризуют возникший по различным причинам отказ информационной системы — приводят к нарушению свойства информационной системы выполнять требуемые функции обработки информации в безопасном режиме.

Это позволяет строить для расчетов соответствующих параметров и характеристик интегрированной информационно-эксплуатационной безопасности единую математическую модель, в которую должны включаться и состояния отказов безопасности, и состояния отказов надежности.

Проиллюстрируем предлагаемый подход к моделированию простым примером. Промоделируем надежные параметры и характеристики уже интегрированной информационно-эксплуатационной безопасности, в предположении, что информационную систему составляет лишь один элемент, который отказывает с интенсивностью λ_n , восстанавливается с интенсивностью μ_n (параметры надежности). Пусть для данного элемента имеет место только одна угроза атак, создаваемая одной угрозой уязвимостей реализации (одного типа), возникающих в системе с интенсивностью λ_6 , устраняемых с интенсивностью μ_6 (надежные параметры безопасности угрозы уязвимостей).

Определим следующее понятие. Под *надежностью интегрированной информационно-эксплуатационной безопасности* будем понимать свойство ин-

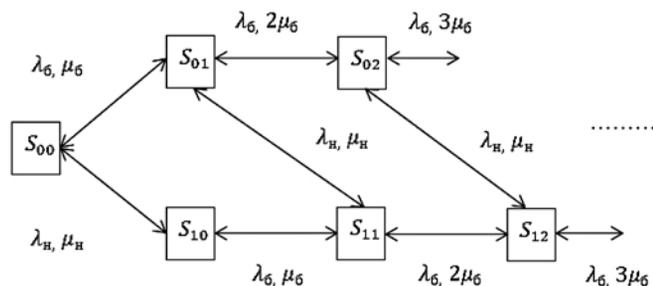


Рис. 5. Размеченный граф системы состояний случайного процесса для модели угрозы интегрированной информационно-эксплуатационной безопасности

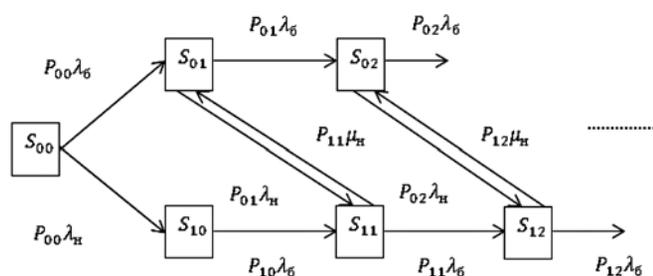


Рис. 6. Модель вероятностного разрезания входных потоков случайных событий

формационной системы сохранять во времени в установленных пределах значения всех характеристик информационной безопасности, определяющих способность выполнять информационной системой требуемые функции обработки информации в безопасном режиме, в условиях отказов элементов информационной системы.

В данных предположениях граф системы состояний случайного процесса для модели угрозы интегрированной информационно-эксплуатационной безопасности принимает вид, приведенный на рис. 5, где состояния системы обозначены S_{ij} , где i — отказ надежности (принимает значение 0 — система работоспособна либо 1, поскольку отказаться уже отказавшая система не может), j — число одновременно возникших в системе реальных угроз уязвимостей (в нашем примере реальных угроз атак, создаваемых одной угрозой уязвимостей).

Отметим, что состояние S_{10} характеризует отказ системы, при этом из данного состояния возможны соответствующие последующие переходы для входного потока случайных событий возникновения реальных угроз уязвимостей, поскольку уязвимости могут возникать (они могут выявляться для соответствующих программных средств, используемых в информационной системе) и при отказавшем элементе, что характеризуется состояниями $S_{1,j>0}$.

Применительно к построенной подобным образом модели по полной аналогии с тем, как это делалось ранее, можно определить значения надежных параметров и характеристик уже интегрированной информационно-эксплуатационной безопасности информационной системы. В частности, вероятность готовности к безопасной эксплуатации информационной системы в условиях отказов элементов (как доля времени ее эксплуатации в безопасном состоянии), модель которой приведена на рис. 5, определяется как P_{00} . Это уже комплексная характеристика, позволяющая учитывать и отказы элементов, в том числе при их резервировании, и отказы безопасности информационной системы.

Совокупность состояний $S_{0,j>0}$ — сумма вероятностей пребывания в них системы, определяет долю времени отказа безопасности информационной системы (возникла реальная угроза) при работоспособности системы. Совокупность состояний $S_{1,j>0}$ — сумма вероятностей пребывания в них системы, определяет долю времени, в течение которого отказавший элемент восстанавливается, доступ к обрабатываемой информации при этом невозможен.

Отметим, что в данном случае имеет место существенное отличие в построении модели вероятностного разрежения входных потоков случайных событий. Модель, построенная для исходной марковской модели (см. рис. 5), приведена на рис. 6.

Отличие состоит в том, что поток восстановлений отказавшего элемента, имеющий интенсив-

ность μ_H , может интерпретироваться как поток возникновения в системе реальных угроз уязвимостей, возникающих во время восстановления отказавшего элемента (что описывается переходами из состояний $S_{1,j>0}$ в соответствующие состояния $S_{0,j>0}$ (см. рис. 6).

Поскольку любое состояние $S_{i \neq 0, j \neq 0}$ может быть интерпретировано как отказ по одной из двух причин (либо отказ элемента, либо возникновение реальной угрозы атаки) готовности к безопасной эксплуатации информационной системы в условиях отказов элементов, комплексный надежностный параметр "интенсивность отказов информационно-эксплуатационной безопасности" информационной системы для модели, приведенной на рис. 6, может быть определен следующим образом:

$$\lambda_{иб} = P_{00}(\lambda_H + \lambda_0).$$

Соответственно надежностный параметр "интенсивность восстановлений интегрированной безопасности" информационной системы

$$\mu_{иб} = \frac{P_{00}(\lambda_H + \lambda_0)}{1 - P_{00}}.$$

Соответствующим образом рассчитываются и временные надежностные характеристики интегрированной информационно-эксплуатационной безопасности информационной системы — среднее время между отказами интегрированной безопасности информационной системы $T_{м0иб}$, среднее время наработки на отказ интегрированной безопасности (восстанавливаемая система) $T_{0иб}$, среднее время восстановления интегрированной безопасности $T_{0виб}$:

$$T_{м0иб} = \frac{1}{\lambda_{иб}}; T_{0иб} = T_{м0иб} - T_{0виб}; T_{0виб} = \frac{1}{\mu_{ибс}}.$$

Очевидно, что событие "возникновение реальной угрозы уязвимостей" никак не влияет на отказоустойчивость элемента, а вот отказ элемента, наоборот, влияет на события возникновения в системе реальных угроз уязвимостей. Это обусловливается тем, что в процессе восстановления отказавшего элемента уязвимость может возникнуть и быть устранена, т. е. она не станет реальной для информационной системы, либо частично она может устраняться при отказавшем и восстанавливаемом элементе (рис. 6). То есть надежностные параметры и характеристики безопасности могут быть уточнены с учетом отказов и восстановлений элементов.

Для их моделирования на модели, приведенной на рис. 6, объединим состояния $S_{0,j>0}$ в одно состояние $S_{00,j>0}$ и состояния $S_{1,j>0}$ в одно состояние $S_{01,j>0}$, в результате чего получим модель вероятностного разрежения входных потоков случайных событий с объединенными состояниями, представленную на рис. 7.

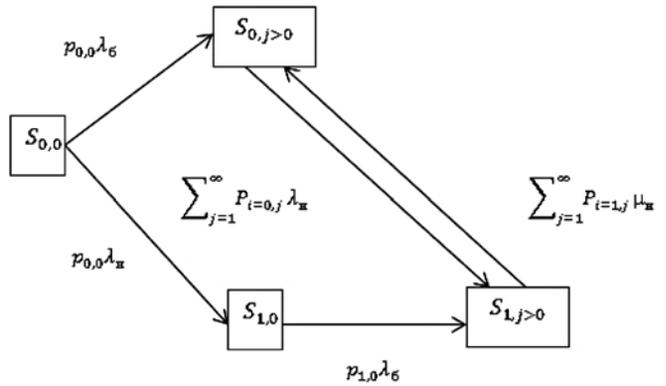


Рис. 7. Модель вероятностного разрежения входных потоков случайных событий с объединенными состояниями

Через $P_{i=0,j}$ на рис. 7 обозначены вероятности нахождения системы в состояниях $S_{i=0,j>0}$, через $P_{i=1,j}$ — в состояниях $S_{i=1,j>0}$ (см. рис. 6).

Заметим, что подобное объединение состояний корректно, поскольку из объединяемых состояний под воздействием одного и того же потока входных событий реализуются переходы в одно и то же, в данном случае объединенное, состояние.

Данная модель позволяет определить надежностные параметры и характеристики безопасности с учетом влияния на их значения отказов элементов.

Если, как и ранее, обозначить параметры отказов (в системе присутствует, по крайней мере, одна реальная угроза атаки) и восстановлений (все реальные угрозы атак в системе устранены) безопасности информационной системы в отношении угрозы атаки как λ_0 и μ_H , то для модели, приведенной на рис. 7, они могут быть рассчитаны следующим образом:

$$\lambda_0 = P_{00}\lambda_0 + \sum_{j=1}^{\infty} P_{i=1,j}\mu_H - \sum_{j=1}^{\infty} P_{i=0,j}\lambda_H;$$

$$\mu_H = \frac{P_{00}\lambda_0 + \sum_{j=1}^{\infty} P_{i=1,j}\mu_H - \sum_{j=1}^{\infty} P_{i=0,j}\lambda_H}{\sum_{j=1}^{\infty} P_{i=1,j}},$$

где параметр μ_H рассчитывается с учетом того, что безопасность системы, нарушаемая с интенсивностью

$$\lambda_0, \text{ восстанавливается за долю времени } \sum_{j=1}^{\infty} P_{i=0,j}$$

(это доля времени нахождения системы в объединенном состоянии $S_{00,j>0}$).

Все сказанное относится и к моделированию эксплуатационных характеристик и параметров интегрированной информационно-эксплуатационной безопасности информационной системы — к моделированию реализации угрозы.

Под эксплуатационной интегрированной информационно-эксплуатационной безопасностью будем

понимать свойство информационной системы сохранять во времени в установленных пределах значения всех характеристик безопасности информации, определяющих способность выполнять информационной системой требуемые функции обработки информации в безопасном режиме, в условиях отказов элементов информационной системы и реализации атак на информационную систему потенциальным нарушителем в процессе ее эксплуатации.

Моделирование эксплуатационных параметров и характеристик интегрированной информационно-эксплуатационной безопасности информационных систем, а также проектирование для них систем защиты информации реализуются с использованием соответствующих подходов и методов моделирования, рассмотренных в работе [2].

Принципиальным отличием модели, как и в предыдущем случае, при этом будет то, что во время отказа элемента системы могут возникать реальные угрозы уязвимостей, в том числе их совокупности, создающие реальные угрозы атак. В результате этого при вводе в эксплуатацию восстановленного элемента на него сразу же, с использованием присутствующих реальных угроз уязвимостей, возникших в процессе восстановления элемента (без необходимости возникновения в системе новых реальных угроз уязвимостей после ввода элемента в эксплуатацию), может быть реализована атака потенциальным нарушителем.

Учет данной особенности моделирования проиллюстрирован на примере, представленном на рис. 8, где опять же предполагается, что информационную систему составляет лишь один элемент, который отказывает с интенсивностью λ_H , восстанавливается с интенсивностью μ_H (параметры надежности), при этом для данного элемента имеет место только одна угроза атаки, создаваемой одной угрозой уязвимостей реализации (одного типа), возникающих в системе с интенсивностью λ_G , устраняемых с интенсивностью μ_G (надежные параметры безопасности угрозы уязвимостей).

В данном случае, как отмечали ранее, уже моделируется система с потерями входных событий.

На рис. 8 использованы следующие обозначения: P_G — характеристика безопасности — вероятности реализации (готовности к реализации) потенциальным нарушителем реальной угрозы уязвимостей (в данном примере — реальной угрозы атаки, создаваемой угрозами уязвимостей одного типа); S_{00} — состояние, характеризующее работоспособное состояние системы и отсутствие в системе реализации потенциальным нарушителем реальных угроз уязвимостей; S_{01} — потенциальным нарушителем реализована реальная угроза уязвимостей; S_{10} — отказ работоспособности (выход из строя) элемента (системы); $S_{1,j>0}^*$ — возникновение в системе одновременно j реальных угроз уязвимос-

тей при выходе из строя элемента системы при реализации потенциальным нарушителем реальной угрозы атаки; $S_{1,j>0}$ — возникновение в системе одновременно j реальных угроз уязвимостей при отказавшем элементе (реальные угрозы атаки при этом потенциальным нарушителем не реализуются).

Особенность моделирования в данном случае состоит в моделировании переходов из состояний $S_{1,j>0}$ и из состояний $S_{1,j>0}^*$, характеризующих возникновение в системе одновременно j реальных угроз уязвимостей при отказавшем элементе. При восстановлении работоспособности элемента (при вводе его в эксплуатацию) с интенсивностью μ_H переход из этих состояний можно интерпретировать как возникновение в системе одновременно j реальных угроз уязвимостей, которыми может воспользоваться потенциальный нарушитель для реализации атаки.

Для оценки этого события используется геометрическое распределение характеристики P_G [2]. При этом учитываем, что хотя бы одна из одновременно возникших подобным образом j реальных угроз уязвимостей будет реализована потенциальным нарушителем (если будет, в противном случае потеряны — это обуславливает переход из состоя-

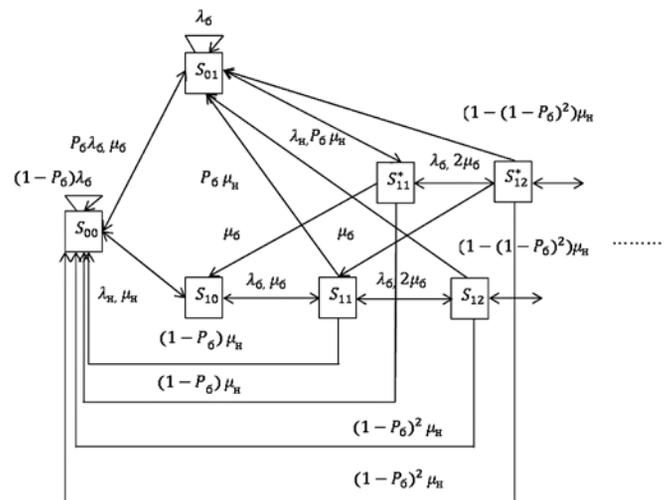


Рис. 8. Размеченный граф системы состояний случайного процесса для модели реализации угрозы интегрированной информационно-эксплуатационной безопасности информационной системы

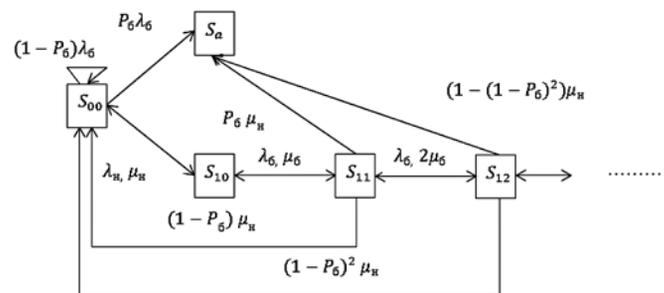


Рис. 9. Размеченный граф системы состояний случайного процесса для модели реализации угрозы безопасности информации

ний $S_{1,j} > 0$, $S_{1,j}^* > 0$ в состояние S_{01} , которое характеризует реализацию потенциальным нарушителем реальной угрозы атаки.

По полной аналогии можно построить и модель реализации угрозы интегрированной информационно-эксплуатационной безопасности информации — реализации потенциальным нарушителем атаки, направленной на нарушение конфиденциальности обрабатываемой информации, с использованием соответствующего поглощающего состояния S_a (рис. 9). В данном случае моделируется не отказ безопасности информационной системы, являющейся восстанавливаемой, а отказ безопасности обрабатываемой в ней информации. При этом, естественно, в отношении нарушения конфиденциальности информации система уже моделируется как не восстанавливаемая [2].

Исходя из всего сказанного, можем сделать вывод, что изложенные в работе подходы к математическому моделированию могут применяться для проектирования отказоустойчивых защищенных информационных систем.

Заключение

В заключение отметим, что моделирование отказоустойчивых защищенных информационных систем — это самостоятельная, достаточно сложная задача [9—11]. Авторы ни в коей мере не претендуют на то, что все необходимые исследования в рамках сформулированной и рассматриваемой задачи моделирования полностью ими решены в объеме данной статьи. Вместе с тем, в этой работе определены основы решения данной задачи — собственно сформулирована задача моделирования,

определен используемый для моделирования математический аппарат, предложены основные подходы к моделированию отказоустойчивых защищенных информационных систем, которые могут использоваться при их проектировании.

Список литературы

1. Щеглов К. А., Щеглов А. Ю. Вопросы корректности и универсальности подхода к моделированию надежностных параметров и характеристик // Информационные технологии. 2016. Т. 22, № 11. С. 854—862.
2. Щеглов К. А., Щеглов А. Ю. Эксплуатационная безопасность. Моделирование реализации угроз атак потенциальным нарушителем // Информационные технологии. 2017. Т. 23, № 1. С. 34—40.
3. Щеглов К. А., Щеглов А. Ю. Возможности методов резервирования для повышения уровня интегрированной информационно-эксплуатационной безопасности современных информационных систем // Информационные технологии. 2015. Т. 21, № 7. С. 521—527.
4. Щеглов К. А., Щеглов А. Ю. Задачи и методы резервирования в области информационной безопасности // Известия высших учебных заведений. Приборостроение. 2015. Т. 58, № 7. С. 507—514.
5. Щеглов К. А., Щеглов А. Ю. Математические модели эксплуатационной информационной безопасности // Вопросы защиты информации. 2014. Вып. 106, № 3. С. 52—65.
6. Щеглов К. А., Щеглов А. Ю. Интерпретация и моделирование угрозы атаки на информационную систему. Часть 1. Моделирование угрозы уязвимости и интерпретация угрозы атаки // Информационные технологии. 2015. Т. 21, № 12. С. 930—940.
7. Щеглов К. А., Щеглов А. Ю. Сложность реализации угрозы атаки и математическая модель потенциального нарушителя // Информационные технологии. 2017. Т. 23, № 2. С. 142—150.
8. Половко А. М., Гуров С. В. Основы теории надежности. СПб.: БХВ-Петербург, 2006.
9. Черкесов Г. Н. Надежность аппаратно-программных комплексов. СПб.: Питер, 2005.
10. Шубинский И. Б. Функциональная надежность информационных систем: методы анализа. М.: Ульяновск: Надежность, 2012. 295 с.
11. Богатырев В. А. Информационные системы и технологии. Теория надежности. М.: Юрайт, 2016.

K. A. Shcheglov, Graduate Student, A. Yu. Shcheglov, Professor, e-mail: info@npp-itb.spb.ru, University ITMO, St. Petersburg, Russia

Integrated Information-Operational Security Parameters and Characteristics Modeling Approach

We did research modern secure fault-tolerant information systems mathematical modeling problems in general. We formulated modeling problem, defined mathematical mechanism used for modeling, suggested basic integrated information-operational security parameters and characteristics modeling approaches — characteristics and parameters informational systems reliability and security. Such parameters and characteristics are determined on single complex model which allows to calculate their interference. Suggested modeling approaches can be applied while building failover secure informational systems.

Keywords: informational system security, information security, attack threat, vulnerability threat, informational system element failure, reliability, redundancy, mathematical modeling, comprehensive assessment, Markov model, integrated security

References

1. Shcheglov K. A., Shcheglov A. Yu. Voprosy korrektnosti i universal'nosti podhoda k modelirovaniyu nadezhnostnykh parametrov u harakteristic, *Informacionnye tehnologii*, 2016, vol. 22, no. 11, pp. 854—862.

2. Shcheglov K. A., Shcheglov A. Yu. Ekspluatatsionnaya bezopasnost'. Modelirovanie realizatsii ugroz atak potencial'nym narushitelem, *Informacionnye tehnologii*, 2017, vol. 23, no. 1, pp. 34—40.

3. Shcheglov K. A., Shcheglov A. Yu. Vozmozhnosti metodov rezervirovaniya dlya povysheniya urovnya integrirovannoy informa-

ционно-эксплуатационной безопасности современных информационных систем, *Информационные технологии*, 2015, vol. 21, no. 7, pp. 521–527.

4. **Shcheglov K. A., Shcheglov A. Yu.** Zadachi i metody rezervirovaniya v oblasti informacionnoy bezopasnosti, *Izvestiya vysshih uchebnyh zavedenij. Priborostroenie*, 2015, vol. 58, no. 7, pp. 507–514.

5. **Shcheglov K. A., Shcheglov A. Yu.** Matematicheskie modeli ehkspluatacionnoy informacionnoy bezopasnosti, *Voprosy zashchity informacii*, 2014, vyp. 106, no. 3, pp. 52–65.

6. **Shcheglov K. A., Shcheglov A. Yu.** Interpretaciya i modelirovanie ugrozy ataki na informacionnyuyu sistemu. Chast' 1. Modelirovanie ugrozy uyazvimosti i interpretaciya ugrozy ataki, *Informacionnye tehnologii*, 2015, vol. 21, no. 12, pp. 930–940.

7. **Shcheglov K. A., Shcheglov A. Yu.** Slozhnost' realizacii ugrozy ataki i matematicheskaya model' potencial'nogo narushitelya, *Informacionnye tehnologii*, 2017, vol. 23, no. 2, pp. 142–150.

8. **Polovko A. M., Gurov S. V.** *Osnovy teorii nadezhnosti*. SPb.: BHV-Peterburg. 2006.

9. **Cherkesov G. N.** *Nadezhnost' apparatno-programmnyh kompleksov*. SPb.: Piter. 2005.

10. **Shubinskij I. B.** *Funkcional'naya nadezhnost' informacionnyh sistem: metody analiza*. Moscow, Nadezhnost, 2012.

11. **Bogatyrev V. A.** *Informacionnye sistemy i tehnologii. Teoriya nadezhnosti*. Moscow, Yurajt, 2016.

УДК 004.7

С. М. Авдошин, канд. техн. наук, проф., руководитель Департамента программной инженерии факультета компьютерных наук НИУ ВШЭ, e-mail: savdoshin@hse.ru,

А. В. Лазаренко, студент, e-mail: avlazarenko@edu.hse.ru,

Национальный исследовательский университет "Высшая школа экономики" (НИУ ВШЭ)

Пиринговые ботнеты

Компьютерные сети, состоящие из зараженных машин, так называемые ботнеты, являются серьезной угрозой для мирового сообщества. Ботнеты являются прекрасным инструментом для организации распределенных атак отказа в обслуживании, рассылки спама, фишинговых писем и кражи конфиденциальной информации. Наиболее простым способом создания ботнета является конструирование центрального сервера и клиентских программ (ботов), поддерживающих с ним связь. Ботнеты такого типа называются традиционными. Главной уязвимостью традиционных ботнетов является центральный сервер. Для того чтобы ее устранить, были спроектированы пиринговые архитектуры. В работе представлен обзор пиринговых ботнетов. Рассмотрена архитектура ботнетов и жизненный цикл ботов. Описаны атаки, организуемые с использованием зомби-сетей.

Ключевые слова: P2P, ботнет, зомби-сеть, пиринговая сеть, вредоносное программное обеспечение

Введение

Бот (сокращение от слова робот) — программа, установленная на компьютер пользователя (жертвы), с целью предоставить возможность злоумышленнику выполнять действия с использованием ресурсов инфицированного компьютера. Ботом мы будем также называть зараженный программой — ботом — компьютер.

Ботнет, или зомби-сеть, — сеть зараженных ботами компьютеров (хостов), подконтрольных злоумышленнику (ботмастеру) и предоставляющих сервисы формата "клиент-сервис" в режиме сервера.

Ботнеты эволюционировали из появившихся в 1998 г. троянских программ бэкдоров, которые устанавливались на компьютеры и работали без уведомления и разрешения пользователей. На основе стека протоколов TCP/IP злоумышленник сам осуществлял соединение с инфицированными компьютерами и мог одновременно управлять сразу несколькими ботами, работающими в режиме сервера. Следующее поколение ботов, появившееся в начале 2000-х годов, взаимодействовало уже со злоумышленником самостоятельно, выходя с ним

на связь по каналу IRC (Internet Relay Chat). Злоумышленник, появляясь онлайн, видел список доступных ботов в сети и отсылал команды либо всем, либо конкретным ботам. Таким образом, был реализован ботнет с механизмом централизованного управления C & C (Command & Control Centre). Кроме IRC-ориентированных более редко встречаются также IM (Instant Messaging)-ориентированные ботнеты, использующие каналы IM служб, таких как AIM, MSN, ICQ и др. Следующий шаг эволюции ботнетов — создание веб-ориентированных ботнетов. Это решение удобно своей простотой для неискушенного пользователя, поэтому весьма популярно сегодня. Множеством ботов теперь можно управлять с любого устройства, поддерживающего доступ в Интернет, в том числе и с мобильного телефона.

На сегодняшний день ботнеты представляют собой большую угрозу для мирового сообщества [1], поскольку зомби-сети, объединяющие ресурсы сотен тысяч и даже миллионов инфицированных компьютеров, являются грозным оружием в руках злоумышленников и источником нелегальной заработка.

Ботнеты являются эффективным инструментом для:

- рассылки спама (по данным компании Cisco 95 % спама в Интернете рассылается с использованием ботнетов [2]);
- фишинга (использование ботов в качестве прокси-серверов позволяет скрыть реальный адрес веб-сервера фишера, для того чтобы противостоять попаданию адресов фишинговых страниц в черные списки);
- проведения распределенных атак типа "отказ в обслуживании" (DDoS-атаки с объемом трафика несколько гигабитов в секунду не являются редкостью [2]);
- продажи и аренды готовых ботнетов злоумышленникам (одна из разновидностей киберпреступного бизнеса);
- хищения конфиденциальной информации, в том числе персональных данных с компьютеров пользователей;
- кражи денег с мобильных и банковских счетов;
- майнинга криптовалют (например, Bitcoin (BTC), Litecoin (LTC) и других Namecoin — пиринговых электронных платежных систем, использующих одноименную криптовалюту).

Развитие Интернета и появление новых технологий стимулирует эволюцию ботнетов. Появляются ботнеты, в основе которых лежат новые сетевые архитектуры, например, пиринговые с архитектурой P2P и децентрализованным механизмом контроля и управления. Таким образом, актуальность исследования ботнетов вызвана необходимостью разработки методов противостояния киберугрозам, исходящим от ботнетов новых типов.

В ботнетах традиционного типа на сервере может быть установлено, например, программное обеспечение, поддерживающее протокол прикладного уровня для обмена сообщениями в режиме реального времени. Ботнеты такого типа зачастую характеризуются большим объемом трафика, относящимся к различным информационным ресурсам: веб-сайтам, Интернет-сервисам и т.д. Как правило, боты в ботнетах традиционной архитектуры посылают запросы к центральному серверу под управлением ботмастера. На таком сервере развернута панель администрирования ботнета.

Централизованный механизм C & C является уязвимым местом ботнета, поэтому представляет интерес для антихакеров (специалистов по информационной безопасности) [3] по следующим причинам:

- при распространении команд ботам через централизованный механизм C & C злоумышленник подвержен риску, поскольку оставляет большое число следов. Если антихакеры получают контроль над ботом, то они могут понять структуру команд ботнета и поставить весь ботнет на мониторинг;

- абсолютно все боты устанавливают соединение с центральным сервером, поэтому на нем есть актуальная информация о всех зараженных машинах. При получении контроля над административной панелью антихакеры могут получить полный список IP-адресов зараженных устройств;
- логичным способом разрушения всего ботнета является нарушение работоспособности центрального сервера любым доступным способом. Без C & C ботнет перестанет функционировать.

Для защиты ботнетов часто используют такие методы, как, например, обфускация, мутация и шифрование исполняемого кода. В процессе обфускации исполняемый код бота приводится к виду, сохраняющему его функциональность, но затрудняющему реверс-инжиниринг, анализ и понимание алгоритмов его работы. Мутация кода бота и его шифрование являются защитой от сигнатурных антивирусов.

Разрушение пирингового ботнета является далеко не тривиальной задачей. Одна из причин — отсутствие одного центрального сервера.

Сегодня существует большое число мобильных ботнетов традиционной архитектуры на платформе Android [4]. Однако до сих пор нет мобильных ботнетов пиринговых архитектур, поскольку смартфоны могут быть отключены от сети в любой момент времени, и связность сети из устройств будет существенно нарушена. Так как в пиринговых архитектурах команды распространяются через все устройства, постоянное подключение к сети является очень важным условием для эффективного функционирования ботнета. Ботнеты, состоящие из персональных компьютеров, не обременены такой проблемой, поскольку рабочие станции могут находиться в сети на протяжении всего рабочего дня.

С ростом популярности Интернета вещей стали появляться ботнеты, ботами в которых являются камеры наружного наблюдения, лампочки и любые другие устройства, имеющие хоть какую-нибудь операционную систему и доступ к Интернету [5]. В ноябре 2016 г. был инцидент с масштабной DDoS-атакой на ряд крупных российских банков с использованием IoT-устройств [6].

По оценкам Gartner к 2018 г. в мире будет около 6 млрд подключенных к глобальной сети IoT-устройств [7]. Пиринговые архитектуры прекрасно подходят для IoT-ботнетов, что может быть вполне вероятным вектором развития последующих разработок.

Ниже приведены используемые в работе термины и сокращения.

AIM (от англ. AOL Instant Messenger) — программа мгновенного обмена сообщениями, предлагаемая фирмой AOL (America On Line), не поддерживается с 2015 г.

Android — операционная система для смартфонов, Интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок и прочих устройств.

C & C (от англ. Command and Control) — контроль и управление.

DDoS (от англ. Distributed Denial of Service) — хакерская атака на вычислительную систему в целях довести ее до отказа, т. е. создание таких условий, при которых добросовестные пользователи системы не могут получить доступ к предоставляемым системным ресурсам.

DDSR graph (от англ. Dynamic Distributed Self Repairing graph) — динамический распределенный самозалечивающийся граф.

DGA (от англ. Domain Name Generation Algorithm) — алгоритм генерации доменных имен.

DNS (от англ. Domain Name System — система доменных имен) — компьютерная распределенная система для получения информации о доменах.

DHT (от англ. Distributed Hash Table) — класс децентрализованных распределенных систем поискового сервиса, работающего подобно хеш-таблице.

Fast-Flux — DNS-техника, используемая ботнетами для сокрытия доставки фишинга и вредоносного ПО за постоянно меняющейся сетью из скомпрометированных узлов, выполняющих роль прокси.

FreeNet — одноранговая сеть, предназначенная для децентрализованного распределенного хранения данных без возможности их цензуры.

GET-запрос — самый распространенный вид HTTP-запроса, с помощью которого происходит запрос браузером любого файла веб-сервера.

HTTP (от англ. HyperText Transfer Protocol) — протокол прикладного уровня передачи данных.

ICQ — бесплатный кроссплатформенный мессенджер, позволяющий обмениваться сообщениями.

IM (от англ. Instant Messaging) — службы мгновенных сообщений (Instant Messaging Service, IMS).

IP-адрес — уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP.

IRC (от англ. Internet Relay Chat) — протокол прикладного уровня для обмена сообщениями в режиме реального времени.

IoT (от англ. Internet of Things) — концепция вычислительной сети физических предметов, оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой.

Kademlia — оверлейная пиринговая сеть над Интернетом, в которой узлы обозначаются специальными номерами ("Node ID") и хранят данные посредством распределенных хеш-таблиц (DHT). Протокол Kademlia определяет структуру сети, регулирующей связь между узлами, и способ обмена информацией в ней. Узлы сети, работающей по протоколу Kademlia, общаются между собой по протоколу транспортного уровня UDP.

MSN (от англ. Microsoft Network) — клиент службы мгновенных сообщений MSN Messenger.

NAT (от англ. Network Address Translation) — механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.

NTP (от англ. Network Time Protocol) — сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью.

Overnet — децентрализованная пиринговая сеть, используемая для хранения больших файлов, протокол которой является дальнейшим развитием сетевого протокола Kademlia в сторону полной децентрализации.

Pull — технология сетевой коммуникации, при которой первоначальный запрос данных проводит клиентом, а ответ порождается сервером.

Push — технология распространения информации (контента) в Интернете, при котором данные поступают от поставщика к пользователю на основе установленных параметров. При этом пользователь либо отвергает, либо принимает данные.

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

TCP/IP — (от англ. Transmission Control Protocol (TCP) и Internet Protocol (IP)) набор сетевых протоколов передачи данных, используемых в сети Интернет.

TOR — (от англ. The Onion Router) — свободное и открытое программное обеспечение для реализации второго поколения так называемой луковой маршрутизации. Это система прокси-серверов, позволяющая устанавливать анонимное сетевое соединение, защищенное от прослушивания.

UDP (англ. User Datagram Protocol) — протокол пользовательских датаграмм — один из ключевых элементов TCP/IP, набора сетевых протоколов, используемых в сети Интернет.

URL (от англ. Uniform Resource Locator) — единообразный локатор (определитель местонахождения) ресурса.

Бот — зараженный программой-ботом компьютер.

Ботнет — сеть зараженных ботами компьютеров (хостов), предоставляющих сервисы формата "клиент-сервис" в режиме сервера и подконтрольных злоумышленнику (ботмастеру).

Бэкдор (от англ. back door — "черный ход") — дополнительная вредоносная функциональность программы или самостоятельная программа, установленная злоумышленником на компьютере жертвы в целях несанкционированного удаленного доступа к компьютеру.

Кейлоггер (от англ. Keylogger) — программное обеспечение или аппаратное устройство, регистри-

рующее различные действия пользователя — нажатия клавиш на клавиатуре компьютера, движения и нажатия клавиш мыши и т. д.

Майнинг (от англ. mining) — деятельность по поддержанию распределенной платформы и созданию новых блоков с возможностью получить вознаграждение в форме эмитированной валюты и комиссионных сборов в различных криптовалютах.

Обфускация (от англ. obfuscation) — метод защиты программного кода, который позволяет усложнить процесс реверсивной инженерии кода защищаемого программного продукта.

Пиринговая сеть (от англ. Peer-to-peer) — оверлейная компьютерная сеть, основанная на равноправии участников.

Прокси-сервер (от англ. proxy) — сервер (комплекс программ) в компьютерных сетях, позволяющий клиентам выполнять косвенные запросы к другим сетевым службам.

Файрвол (от англ. Firewall) — технологический барьер, предназначенный для предотвращения несанкционированного или нежелательного сообщения между компьютерными сетями или хостами.

Фишинг (от англ. Phishing) — вид Интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей — логинам и паролям.

Жизненный цикл пирингового ботнета

Жизненный цикл пирингового ботнета состоит из трех частей: инфицирование, формирование и ожидание поступления команд [8].

На первой стадии ботмастеру необходимо заполнить максимальное число ботов. Для достижения этой цели существует множество сценариев. Ботнет может использовать существующую пиринговую сеть, например, Gnutella [9]. При использовании такой сети необходимо скомпрометировать компьютеры, являющиеся ретрансляторами в сети. Если существующая сеть не используется, то подходят любые другие традиционные способы заражения устройств: электронная почта (фишинг), мгновенные сообщения, файлообменники и прочие средства, предоставляющие возможность распространять вредоносное программное обеспечение.

На второй стадии ботмастеру необходимо сформировать из зараженных устройств (ботов) ботнет (связанную сеть). Под формированием ботнета следует понимать процедуру присоединения новых устройств к ботнету и процесс установления контакта новоприбывших ботов с остальными узлами сети.

На третьей стадии боты ожидают инструкций от ботмастера. Ключевым инструментом является C & S механизм, поскольку от него напрямую зависит эффективность ботнета. Этот механизм целиком и полностью определяет коммуникационную топологию ботнета. В целом, механизм C & S бывает двух типов: pull и push. Механизм C & S типа

pull работает по принципу публикация/подписки на команды. Боты получают команды из того места, в котором ботмастер их публикует. Механизм C & S типа push работает по принципу проталкивания команды. Боты пассивно ожидают поступления команд и пересылают полученные команды другим узлам.

Архитектуры пиринговых P2P ботнетов

Для пиринговых ботнетов чаще всего используются три разновидности архитектур: модерируемая, чистая и гибридная [8]. Все эти разновидности основаны на архитектурах существующих пиринговых сетей [10].

Модерируемая архитектура использует клиент-серверный подход для управления операциями. Некоторые узлы ботнета подсоединяются к центральному серверу и получают от него команды, после чего распространяют полученные команды последующим ботам.

В *чистой* архитектуре центральный сервер не используется вообще. Все узлы в сети коммуницируют друг с другом напрямую. У сетей с чистой архитектурой есть значительный минус: большое количество лишнего трафика. Среди существующих сетей с чистой архитектурой является сеть FreeNet [11].

Гибридная архитектура является компромиссом между двумя предыдущими архитектурами. В ней используется два типа узлов: обычные и мегаузлы. Именно наличие мегаузлов в подобных архитектурах позволяет ботнетам сочетать в себе свойства модерируемой и чистой архитектуры. Мегаузлы используются в качестве центральных серверов, но только для определенного и ограниченного набора узлов. Мегаузлы взаимодействуют друг с другом, как в чистых пиринговых сетях. Обычные узлы получают команды от мегаузлов точно так же, как это происходит в модерируемой архитектуре.

В таблице приведены примеры ботнетов указанных архитектур, упорядоченных по году их появления. Рассмотрим эти ботнеты более подробно.

Storm — ботнет модерируемой архитектуры, созданный в 2007 г. на основе вредоносной программы Storm Worm [12]. Основной целью ботнета Storm являлась рассылка спама. Благодаря модульной архитектуре вируса он легко перестраивается в исполнителя DDoS-атак. Будучи единожды инфицированным, бот присоединяется к сети, использующей протокол Overnet, работающий поверх пиринговой сети Kademia [13] для распространения информации и обеспечения других ботов необходимыми ресурсами.

Присоединение нового узла является одной из главных проблем пиринговых сетей. Когда такой узел присоединяется, у него нет никакой информации о текущем состоянии сети. При этом большое количество данных должно поставаться в би-

Примеры ботнетов различных архитектур

Название	Назначение	Тип ботнета	Архитектура	Протокол	Год
Storm	Спам/DDoS	Промышленный	Модерируемая	Overnet	2007
Nugache	DDoS	Промышленный	Гибридная	Свой	2007
АНР2Р 1.0	Нет данных	Экспериментальный	Гибридная	Свой	2007
Overbot	Нет данных	Экспериментальный	Чистая	Kademlia	2009
Waledac	Спам	Промышленный	Гибридная	Свой	2009
АНР2Р 2.0	Нет данных	Экспериментальный	Гибридная	Свой	2011
Hlux/Kelihos	Спам/DDoS	Промышленный	Гибридная	Свой	2011
Miner	Майнинг BTC	Промышленный	Модерируемая	Свой	2011
Zeus	DDoS	Промышленный	Гибридная	Свой	2013
OnionBot	Нет данных	Экспериментальный	Гибридная	Свой, на основе Tor	2015

нарном виде, чтобы позволить присоединенному боту взаимодействовать с другими узлами. Если бот подсоединяется к какому-то мегаузлу для получения контактного списка узлов, то обнаружение его IP-адреса и последующая блокировка не составляют никакого труда.

Для решения этой проблемы разработчики Storm используют следующий подход: каждый бот поставлялся с записями о 300 статических ботах в текстовом файле. Записи в файле содержат информацию о хэше бота, IP-адресе, порте и типе бота. Эти боты являются единственными узлами, известными новому боту. Бот подключается к этим узлам, чтобы получить актуальную информацию о топологии сети. Однако бот не может определить внешний IP-адрес, поэтому он посылает специальный запрос для его выяснения.

После того как бот подключается к другим узлам и получает актуальную информацию о топологии сети, он становится готовым либо для рассылки спама, либо для осуществления DDoS-атаки. Процесс получения актуальной информации о составе сети является непрерывным и продолжается на протяжении всего времени существования бота.

Storm бот распространяется посредством сообщений электронной почты. Тело сообщения содержит разнообразный англоязычный текст, который пытался ввести жертву в заблуждение, чтобы она кликнула на вложенную ссылку (фишинг). Для распространения вредоноса используются техники социальной инженерии.

Во время установки вирус сохраняет конфигурационный файл в зараженной системе. В файле содержится закодированная информация об остальных ботах, с которыми программа взаимодействует после установки. Каждый бот идентифицируется с помощью хэша и комбинации из IP-адреса и порта. Этот набор является базовым и необходимым для присоединения к пиринговой сети. Storm синхронизирует системное время зараженной машины с помощью протокола NTP [14].

Для поиска других ботов в сети и получения команд от ботмастера Storm бот использует прото-

кол распределенных хэш-таблиц Overnet [15]. Он стремится собрать в одном месте как можно больше активных узлов.

Nugache — ботнет гибридной архитектуры [16]. Главное назначение этого бота — организация DDoS-атак. Кроме того, бот умеет организовывать удаленный запуск различных команд и имеет базовую функциональность кейлоггера. Каждый бот поставляется с прошитым списком узлов в бинарном формате. Узлы подключаются к сети на основе обмена RSA-ключами и актуальными списками узлов в сети. После того как зашифрованное соединение устанавливается, внутренний протокол ботнета используется для выполнения заданий. Одним из таких заданий является инициация процесса, используемого для определения IP-адреса узла и слушающего порта, а также определяющего, что бот может быть не только ботом-служителем, но и ботом-клиентом. Другим заданием является проверка необходимости обновления программного обеспечения или списка узлов. Проблема ботнета Nugache состоит в необходимости поддержания актуальности списка узлов для переподключения к пиринговой сети.

Все команды в ботнете Nugache распространяются через зашифрованный пиринговый канал между ботами-клиентами и ботами-служителями, которые могут выполнять роль как клиента, так и сервера.

Узлы ботнета Nugache держат не более десяти входящих соединений в любой момент времени и менее десяти исходящих. В результате типичный узел имеет 13...15 соединений с другими ботами в любое время.

Раньше бот использовал фиксированный набор портов и протокол IRC для приема команд от ботмастера. Позднее порты для механизма C & C начали определяться случайно, что уменьшило детектируемость этого ботнета.

АНР2Р 1.0 — продвинутый пиринговый ботнет гибридной архитектуры [17]. Множество ботов этого ботнета делится на две группы (рис. 1). Первая группа состоит из ботов, у которых есть статические IP-адреса, доступные из Интернета, — это

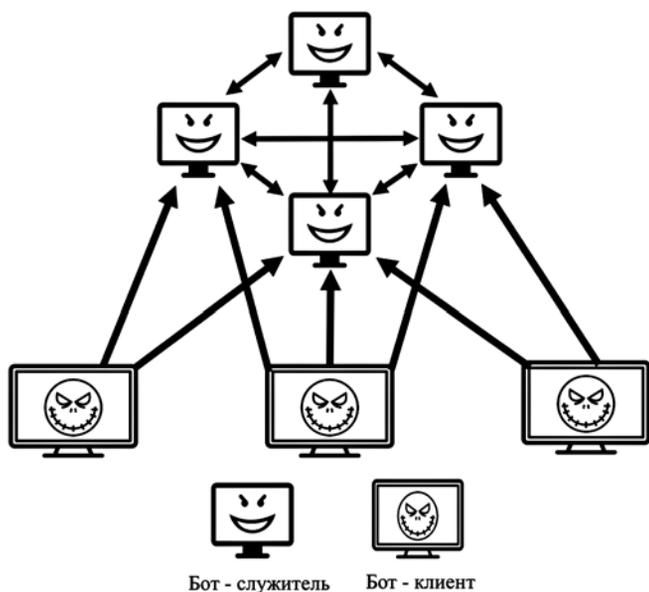


Рис. 1. Структура ботнета ANP2P 1.0

боты-служители. Вторая группа содержит боты с динамическими IP-адресами, боты с приватными IP-адресами и боты за фаерволом, к которым нет доступа из Интернета. Вторая группа ботов называется клиентами, потому что они не принимают входящих соединений, в отличие от ботов первой группы.

На начальной стадии боты поставляются со сформированным списком ботов-служителей. Так как служители не изменяют свой IP-адрес, ботнет становится более стабильным, но и более уязвимым, поскольку существует возможность выведения ботов-служителей из строя в результате атак антихакеров.

Ботмастер может передавать команду через любого бота в ботнете. Как клиенты, так и служители периодически подключаются к ботам-служителям из своих списков, чтобы получать команды от ботмастера. Когда бот получает новую команду, которую он ранее не получал, он незамедлительно передает ее остальным ботам-служителям из списка.

Overbot — ботнет чистой архитектуры, функционирует на основе сети Kademlia [18]. Overbot позволяет скрытно пересылать команды от ботмастера скомпрометированным узлам. Ботнет является устойчивым к компрометации отдельно взятых узлов, а сетевой трафик прячет внутри легитимного трафика сети Kademlia. Узлы ботнета ведут себя точно так же, как легитимные узлы Kademlia. Значительным преимуществом Overbot является тот факт, что ботмастер никогда не передает команду ботам напрямую.

Протокол Overbot использует поисковые запросы в сети, чтобы мотивировать боты запрашивать команду от ботмастера. Каждый бот посылает запросы, идентифицировать которые в качестве ко-

мандных может только ботмастер. На сетевом уровне такие запросы не могут быть отделены от обычных поисковых запросов. Этот факт остается справедливым даже тогда, когда ботмастер контролирует несколько ботов и может просматривать и записывать запросы, которые эти узлы посылают.

Чтобы обработать поисковые запросы, являющиеся командными, Overbot использует модель с публичным ключом. У ботмастера есть своя собственная пара из публичного и приватного ключей. Публичный же ключ ботмастера известен всем скомпрометированным узлам. Для того чтобы взаимодействовать с ботмастером, узлы ботнета шифруют сообщения с помощью публичного ключа ботмастера. Только ботмастер, имеющий приватный ключ, может идентифицировать эти сообщения.

Waledac — ботнет гибридной архитектуры, состоит из нескольких разных слоев [19]. На рис. 2 представлены боты из каждого слоя. В самом нижнем слое находятся боты-спаммеры. Эти боты используются для осуществления спам-компаний. Ключевым свойством, отделяющим ботов-спаммеров от остальных ботов, является отсутствие общедоступного IP-адреса. Это означает, что они могут находиться за NAT-маршрутизатором и не могут быть доступны напрямую из Интернета.

В следующем слое находятся боты-репитеры. Это входные точки других новых ботов, присоединяющихся к сети. Эти точки являются узлами, к которым может обратиться другой действующий бот. По этой причине боты-репитеры имеют публично доступный IP-адрес. Бот-репитер может быть посредником между самым нижним уровнем и бекендом ботнета. Боты-спаммеры подключаются к ботам-репитерам, чтобы получать новые задания от ботмастера или пересылать результаты выполнения предыдущих операций. Эти запросы ретранслируются в следующий слой — бекенд-сервер ботнета. Боты-репитеры, к тому же, могут играть роль Fast-Flux [20] агентов для различных Fast-Flux-доменов Waledac. Это значит, что они могут ретранслировать HTTP-запросы неинфицированных узлов.

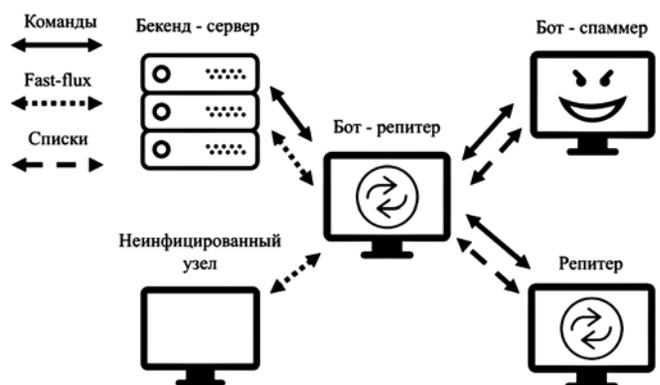


Рис. 2. Схема взаимодействия ботов Waledac

Следующий уровень состоит из бекенд-серверов, отвечающих на запросы как ботов-спаммеров, так и на Fast-Flux-запросы ботов-репитеров. Бекенд-серверы используются еще и для проксирования запросов.

Боты-репитеры и боты-спаммеры постоянно обмениваются списками активных ботов-репитеров. Благодаря этому механизму любой бот в любой момент времени имеет в своем списке активного бота-репитера. Для дополнительной надежности каждый бинарный файл бота имеет список URL адресов, находящихся внутри Fast-Flux-сети Waledac. Эти адреса указывают на другой список активных репитеров. Поэтому если бот не может подключиться ни к одному из 10 репитеров в зашитом списке, он закачивает новый лист с одного из таких адресов. Боты-репитеры обмениваются списками активных в данный момент бекенд-серверов. Этот список подписывается приватным ключом ботмастера, чтобы убедиться, что антихакер не сможет внедрить свой бекенд-сервер в ботнет.

Боты Waledac не содержат встроенных механизмов распространения, следовательно, инфицированные узлы не сканируют локальную сеть в поисках уязвимых систем. Вместо этого боты распространяются с помощью социальной инженерии. Боты-спаммеры иногда получают инструкцию о пересылке электронной почты, содержащей URL, указывающий на актуальную версию Waledac. Для увеличения вероятности инфицирования нового хоста самораспространяющиеся письма маскируются в подарочные открытки.

Инфицирующий цикл следующий: боты-спаммеры регулярно опрашивают ботов-репитеров о новых заданиях. Запросы ретранслируются на один из бекенд-серверов, возвращающий в ответ текущее задание. В этом случае заданием является отсылка спам-сообщений для распространения бота. Неинфицированный узел получает одно из сообщений и проходит по вложенной ссылке, чтобы запросить текущий бинарник Waledac у одного из Fast-Flux-агентов, привязанных к домену. И снова один из бекенд-серверов передает запрос через Fast-Flux-агента обратно запрашивающему тому или иному узлу в зависимости от доступности. Свежий узел затем встает на место бота-репитера или бота-спаммера.

АНР2Р 2.0 — новая версия продвинутого пирингового ботнета гибридной архитектуры [21]. Ботнет состоит из социального веб-сайта и группы ботов. Группа ботов состоит из ботов-служителей и ботов-клиентов. Команды от ботмастера распространяются через социальные веб-сайты. Боты-служители в группах соединяются с социальными сайтами, чтобы получить команду от ботмастера и послать ее ботам-клиентам. Боты-клиенты атакуют цель после того, как получают информацию от бота-служителя. Такой механизм С & С позволяет

скрывать трафик ботмастера в трафике социальных сетей.

Hlux/Kelihos — пиринговый ботнет гибридной архитектуры, очень похожей на Waledac [22]. Встречается под двумя названиями: Kelihos (Microsoft), Hlux (Лаборатория Касперского). Состоит из следующих слоев: боты-контроллеры, боты-маршрутизаторы, боты-рабочие. Боты-контроллеры управляются извне ботнета. Они распространяют команду бота и управляют динамической структурой пиринговой сети. Боты-маршрутизаторы являются зараженными компьютерами с публичными IP-адресами. Они могут запустить другой бот в режиме маршрутизатора, хост прокси сервиса, Fast-Flux-агента и т. д. Боты-рабочие являются зараженными машинами, которые не используются в режиме маршрутизатора. Их применяют для рассылки спама, сбора адресов электронной почты и прочей информации от пользователей.

Каждый бот содержит список из 500 узлов в локальном списке узлов. Этот лист сохраняется в регистрах операционной системы Windows. Лист обновляется, когда бот получает информацию от других ботов.

Данный бот применялся на практике и насчитывал 49 тыс. уникальных IP-адресов.

Miner — узкоспециализированный ботнет [23], разработан в целях майнинга криптовалюты Bitcoin [24]. В процессе инициализации он устанавливает вместе с собой большое количество разнообразного программного обеспечения:

- исполняемый файл, спрятанный в альтернативном потоке данных;
- три Bitcoin-майнера;
- файл с информацией о геолокации для IP-адресов.

Данному боту присвоена следующая сигнатура: Trojan.Win32.Miner.h. Бинарный файл бота имеет несколько слоев обфускации.

В бинарном файле бота содержится список из IP-адресов, которые используются на фазе присоединения бота к пиринговой сети.

Для проверки принадлежности узла к ботнету посылается запрос на порт 62999/tcp. После этого все коммуникации осуществляются через порт 8080/tcp. Если бот хочет получать какую-либо информацию от ботнета, он пересылает GET-запрос на URL другого узла. В ответ он получает запрашиваемую информацию.

Иными словами, любой узел может получить актуальный список доступных узлов, рекурсивно запрашивая информацию об узлах у других ботов.

Zeus — пиринговая версия [25] бота Zeus с гибридной архитектурой, который появился в первый раз в 2007 г.

Централизованная версия Zeus продавалась в качестве Builder-kit, с помощью которого любой пользователь мог собрать приватную версию бота.

Эта опция перестала быть доступной в пиринговой версии Zeus, который основывался на единой и прочной P2P-сети.

Основная сеть была разделена на несколько виртуальных подсетей с помощью под сетевого идентификатора, записанного в каждом бинарном файле бота. В то время как пиринговая сеть Zeus поддерживается и обновляется целиком, подсети независимо контролируются несколькими ботмастерами. По заявлению авторов в сети насчитывается как минимум 200 000 ботов.

Пиринговая сеть Zeus используется для двух целей:

- обмен бинарными и конфигурационными файлами между ботами (в том числе и обновлениями);
- обмен списками прокси-ботов, являющихся выделенными ботами, на которые может быть отправлена украденная информация и получена команда для распространения по сети.

В качестве запасного канала связи в пиринговой сети Zeus используется DGA-алгоритм [26] на случай, если контакт с обычной пиринговой сетью пропадет.

Пиринговая сеть Zeus является иерархической сетью, состоящей из нескольких слоев. В нижнем слое находятся пиринговые боты. Периодически некоторое подмножество ботов этого слоя получает статус прокси-ботов. Эта операция выполняется ботмастерами вручную, с помощью отправки криптографически подписанного сообщения в сеть. Так называемые харвестер-боты используют прокси-боты для получения команд и сброса украденных данных.

Прокси-боты играют роль ретрансляторов между пиринговыми ботами и верхним уровнем иерархии, который мы называем C & C-прокси-слоем. C & C-прокси-слой содержит несколько выделенных обычных HTTP-серверов (не ботов). Эти серверы формируют дополнительный слой между прокси-ботами и настоящим C & C-сервером. Периодически прокси-боты взаимодействуют с C & C-прокси-слоем, чтобы провести обновления их репозитория команд и пересылки украденных данных, собранных от ботов сверху по иерархии.

Наконец, на самом верхнем уровне иерархии находится C & C-слой, который является источником команд и конечной точкой назначения украденных данных. Команды распространяются вниз от C & C-слоя, через C & C-прокси-слой к прокси-ботам, где они подхватываются харвестер-ботами. Аналогично данные, украденные харвестер-ботами, собираются прокси-ботами и периодически передаются вверх до тех пор, пока они не достигнут C & C-слоя.

Каждый бот поставляется с прошитым списком узлов. Этот список содержит IP-адреса, порты и уникальные идентификаторы ботов (до 50 шт.).

Сетевая согласованность поддерживается с помощью push/pull обменного механизма списками ботов. Как правило, ботнет Zeus периодически "пропихивает" обновления списков узлов. Когда бот получает сообщение от другого бота, он добавляет этот бот в локальный список узлов, если список содержит менее 50 узлов. Если боту не хватает узлов, то он начинает их активно запрашивать.

Zeus-бот проверяет работоспособность своих соседей каждые 30 мин. С каждым соседом устанавливается соединение, и соседу дается пять попыток для ответа. Если сосед не отвечает в течение пяти попыток, он помечается умершим и исключается из списка узлов.

OnionBot — ботнет гибридной архитектуры [27], выполняет все свои взаимодействия через сеть Tor [28, 29]. Ботнет осуществляет свою деятельность в соответствии с типичным жизненным циклом пирингового ботнета, однако каждая стадия жизненного цикла ботнета содержит уникальные отличия от стадий жизненного цикла других ботнетов. Например, любая запись в списке узлов относится к .onion адресу узла.

На стадии инфицирования ботнет OnionBot использует комбинированный подход, основанный на прошитом списке узлов и списках узлов, которые он может опрашивать для получения адресов новых ботов.

Разработчики ставили перед собой цель сделать пиринговую самозалечивающуюся сеть, которая поддерживает низкую степень узлов и короткое расстояние между ботами для распространения сообщений. Авторы работы [27] предлагают новую конструкцию динамического, распределенного, самозалечивающего графа (DDSR graph). Они предполагают наличие двух типов C & C-сообщений — от ботов к панели управления и от панели управления к ботам. Боты передают свой адрес на панель управления и устанавливают уникальный ключ, которым делятся с ботмастером на стадии инфицирования. Это позволяет ботмастеру получить прямой доступ к ботам даже после того, как они меняют .onion адрес. Каждый бот генерирует симметричный ключ и сообщает его ботмастеру с помощью шифрования публичным ключом. После обмена ключами боты могут периодически менять свой .onion адрес на основе личного приватного. Однако даже использование сети Tor не гарантирует полной анонимности и скрытности [30, 31].

Заключение

В статье были рассмотрены примеры пиринговых ботнетов, являющихся новым поколением, устойчивым против традиционных методов разрушения ботнетов. Однако, как показывает их анализ, даже такие ботнеты уязвимы к некоторым видам атак [8]:

- детекции — замене процесса инфицирования на ранней стадии;
- мониторингу — внедрению сенсоров в сеть для мониторинга процессов внутри ботнета. В этом случае сенсор является узлом ботнета, что позволяет наблюдать за поведением ботнета в реальном времени;
- разрушению — может осуществляться двумя способами: удаление найденных ботов или уничтожение канала C & C. Становление ботнета очень сильно зависит от фазы инфицирования, на которой боты являются крайне уязвимыми. Если разрушить посевные серверы или боты, находящиеся внутри прошитого листа, можно разрушить новорожденный ботнет;
- помимо этих вариантов можно убирать мегаботы и изолировать куски системы. Например, можно разрушить несколько критически важных компонентов для сети, чтобы разделить один большой ботнет на несколько маленьких и сделать невозможным распространение команды на определенные куски.

Список литературы

1. **Ingdahl W.** Botnets: a Serious Threat to the Web // Spiked [Official website]. URL: <http://www.spiked-online.com/newsite/article/botnets-a-serious-threat-to-the-web/15216#.WCYX8HdePwc> (accessed: 11.10.2016).
2. **Официальный документ Cisco**, Ботнет: новый характер угроз // Cisco Systems, Inc. [Официальный вебсайт]. URL: <http://www.cisco.com/web/RU/downloads/Botnets.pdf> (дата обращения: 11.10.2016).
3. **Vogt R., Aycock J., Jacobson M. J. Jr.** Army of Botnets // University of Calgary [Official website]. URL: <http://pages.epsc.ucalgary.ca/~aycock/papers/ndss07.pdf> (accessed: 11.10.2016).
4. **Grooten M.** A Timeline of Mobile Botnets // Virus Bulletin [Official website]. URL: <https://www.virusbulletin.com/virusbulletin/2015/03/timeline-mobile-botnets/> (accessed: 11.10.2016).
5. **Нефедова М.** Обнаружен ботнет, состоящий из 25000 камер видеонаблюдения // Хакер [Официальный вебсайт]. URL: <https://хакер.ru/2016/06/28/cctv-botnet-2/> (дата обращения: 11.10.2016).
6. **Берг Е.** Пять крупных банков подверглись DDoS-атаке. Главное. // Meduza [Официальный вебсайт]. URL: <https://meduza.io/feature/2016/11/10/pyat-krupnyh-rossiyskih-bankov-podverglis-ddos-atake-glavnoe> (дата обращения: 11.10.2016).
7. **Top Strategic Predictions for 2016 and Beyond: The Future Is a Digital Thing** // Gartner [Official website]. URL: <https://www.gartner.com/doc/3142020?plc=ddf> (accessed: 11.10.2016).
8. **Wang P., Wu L., Aslam B., Zou C. C.** A Systematic Study on Peer-to-Peer Botnets // ACM Digital Library [Official website]. URL: <http://dl.acm.org/citation.cfm?id=1635111> (accessed: 11.10.2016)
9. **Gnutella** // Wikipedia [Official website]. URL: <https://en.wikipedia.org/wiki/Gnutella> (accessed: 11.10.2016)
10. **Backx P., Wauters T., Dhoedt B., Demeester P.** A Comparison of Peer-to-Peer Architectures // CiteSeerX [Official website]. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.9453&rep=rep1&type=pdf> (accessed: 10.11.2016).
11. **Freenet** // Freenet [Official website]. URL: <https://freenet-project.org> (accessed: 10.11.2016).
12. **Holz T., Steiner M., Dahl F., Biersack E., Freiling F.** Measurements and Mitigation of Peer-to-Peer based Botnets: A Case Study on Storm Worm // Usenix [Official website]. URL: https://www.usenix.org/legacy/events/leet08/tech/full_papers/holz/holz_html/index.html (accessed: 10.11.2016).
13. **Overnet** // Wikipedia [Official website]. URL: <https://en.wikipedia.org/wiki/Overnet> (accessed: 10.11.2016).
14. **Network Time Protocol** // Wikipedia [Official website]. URL: https://en.wikipedia.org/wiki/Network_Time_Protocol (accessed: 10.11.2016).
15. **Kademlia** // Wikipedia [Official website]. URL: <https://en.wikipedia.org/wiki/Kademlia> (accessed: 10.11.2016).
16. **Stover S., Dittrich D., Hernandez J., Dietrich S.** Analysis of the Storm and Nugache Trojans: P2P is Here // Usenix [Official website]. URL: <https://www.usenix.org/system/files/login/articles/526-stover.pdf> (accessed: 10.11.2016).
17. **Wang P., Sparks S., Zou C. C.** An Advanced Hybrid Peer-to-Peer Botnet // Usenix [Official website]. URL: https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/wang/wang_html/ (accessed: 10.11.2016).
18. **Starnberger G., Kruegel C., Kirda E.** Overbot — A Botnet Protocol Based on Kademlia // Technische Universität Darmstadt [Official website]. URL: https://www.p2p.tu-darmstadt.de/fileadmin/user_upload/Group_TK/overbot.pdf (accessed: 10.11.2016).
19. **Stock B., Gobel J., Engelberth M., Freiling F. C., Holz T.** Walowdac — Analysis of a Peer-to-Peer Botnet // Academia Sinica [Official website]. URL: http://mmnet.iis.sinica.edu.tw/botnet/file/20110117/20110117_1.pdf (accessed: 10.11.2016).
20. **Fast flux** // Wikipedia [Official website]. URL: https://en.wikipedia.org/wiki/Fast_flux (accessed: 10.11.2016).
21. **Lu T. T., Liao H. Y., Chen M. F.** An Advanced Hybrid P2P Botnet 2.0 // WASET [Official website]. URL: <http://waset.org/publications/12157/an-advanced-hybrid-p2p-botnet-2.0> (accessed: 10.11.2016).
22. **Werner T.** Botnet Shutdown Success Story: How Kaspersky Lab Disabled the Hlux/Kelihos Botnet // Securelist [Official website]. URL: <https://securelist.com/blog/research/31058/botnet-shutdown-success-story-how-kaspersky-lab-disabled-the-hluxkelihos-botnet-15/> (accessed: 11.11.2016)
23. **Werner T.** The Miner Botnet: Bitcoin Mining Goes Peer-To-Peer // Securelist [Official website]. URL: <https://securelist.com/blog/incidents/30863/the-miner-botnet-bitcoin-mining-goes-peer-to-peer-33/> (accessed: 11.11.2016).
24. **Bitcoin** // Wikipedia [Official website]. URL: <https://en.wikipedia.org/wiki/Bitcoin> (accessed: 11.11.2016).
25. **Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus** // SysSec [Official website]. URL: http://www.syssec-project.eu/m/page-media/3/zeus_malware13.pdf (accessed: 11.11.2016).
26. **Domain generation algorithm** // Wikipedia [Official website]. URL: https://en.wikipedia.org/wiki/Domain_generation_algorithm (accessed: 11.11.2016).
27. **Sanatinia A., Noubir G.** OnionBots: Subverting Privacy Infrastructure for Cyber Attacks // Arxiv [Official website]. URL: <https://arxiv.org/pdf/1501.03378.pdf> (accessed: 11.11.2016)
28. **Tor** // Wikipedia [Official website]. URL: [https://en.wikipedia.org/wiki/Tor_\(anonymity_network\)](https://en.wikipedia.org/wiki/Tor_(anonymity_network)) (accessed: 11.11.2016).
29. **Авдошин С. М., Лазаренко А. В.** Технология анонимных сетей // Информационные технологии. 2016. Т. 22, № 4. С. 284—291.
30. **Авдошин С. М., Лазаренко А. В.** Deep Web Users Deanonimization System. Proceedings of the Institute for System Programming, 2016. Vol. 28, is. 3. P. 21—34.
31. **Авдошин С. М., Лазаренко А. В.** Методы деанонимизации пользователей Tor. Информационные технологии, 2016. Т. 22, № 5. С. 362—372.

S. M. Avdoshin, Ph. D., Professor, Head of Software Engineering School,
Faculty of Computer Science, e-mail: savdoshin@hse.ru,
A. V. Lazarenko, Undergraduate Student, Software Engineering School,
Faculty of Computer Science, e-mail: avlazarenko@edu.hse.ru,
National Research University Higher School of Economics (HSE)

Peer-to-Peer Botnets

Huge computer networks composed of infected machines so called botnets are currently the one of the most dangerous threats for global society. Botnets are a brilliant tool for organizing distributed denial of service attacks, sending spam and phishing email, stealing sensitive information and money from various accounts. Hackers create botnets from any kind of electronic devices connected to the Internet. The most widely used devices are: personal computers with windows operating system, smartphones with Android OS, IoT devices with public IP addresses and open vulnerabilities (IPTV cameras, for example). There are a lot of options for botnet hiring on the darknet. It is possible to hire a botnet, buy a builder kit or any other service in hacker-to-hacker manner. The easiest way to build a botnet is a creation of central server for spreading commands and creation of clients (bots) that will connect to this server. Such botnets are so called traditional botnets. Traditional botnets with centralized command and control mechanism are vulnerable to the wide range of attacks. For example, if one particular bot is captured by the security analyst the IP address of command and control server will be compromised with very high probability. Without central command and control server botnet is useless because the malefactor is unable to spread commands over infected network. In order to make botnets more robust and steady the peer-to-peer botnet architecture was introduced. P2P architectures could be divided on 3 main categories: pure, moderated and hybrid. The aim of the current papers is covering peer to peer botnets and its architecture.

Keywords: P2P, botnet, zombie network, peer-to-peer network, malware

References

1. **Ingdahl W.** Botnets: a Serious Threat to the Web, *Spiked [Official website]*, URL: <http://www.spiked-online.com/newsite/article/botnets-a-serious-threat-to-the-web/15216#.WCYX8HdePwc> (accessed: 11.10.2016).
2. **Official'nyj dokument Cisco**, Botnet: novyj karakter ugroz, *Cisco Systems, Inc [Official website]*, URL: <http://www.cisco.com/web/RU/downloads/Botnets.pdf> (accessed: 11.10.2016) (in Russian).
3. **Vogt R., Aycock J., Jacobson M. J. Jr.** Army of Botnets, *University of Calgary [Official website]*, URL: <http://pages.cpsc.ucalgary.ca/~aycock/papers/ndss07.pdf> (accessed: 11.10.2016).
4. **Grooten M.** A Timeline of Mobile Botnets, *Virus Bulletin [Official website]*, URL: <https://www.virusbulletin.com/virusbulletin/2015/03/timeline-mobile-botnets/> (accessed: 11.10.2016).
5. **Nefyodova M.** Obnaruzhen botnet, sostoyashchij iz 25 000 kamer videonablyudeniya, *Haker [Official website]*, URL: <https://xakep.ru/2016/06/28/cctv-botnet-2/> (accessed: 11.10.2016) (in Russian).
6. **Berg E.** Pyat' krupnyh bankov podverglis' DDoS-atake. Glavnoe [Electronic resource], *Meduza [Official website]*, URL: <https://meduza.io/feature/2016/11/10/pyat-krupnyh-rossijskih-bankov-podverglis-ddos-atake-glavnoe> (accessed: 11.10.2016) (in Russian).
7. **Top Strategic Predictions for 2016 and Beyond: The Future Is a Digital Thing**, *Gartner [Official website]*, URL: <https://www.gartner.com/doc/3142020?plc=ddf> (accessed: 11.10.2016).
8. **Wang P., Wu L., Aslam B., Zou C. C.** A Systematic Study on Peer-to-Peer Botnets, *ACM Digital Library [Official website]*, URL: <http://dl.acm.org/citation.cfm?id=1635111> (accessed: 11.10.2016)
9. **Gnutella**, *Wikipedia [Official website]*, URL: <https://en.wikipedia.org/wiki/Gnutella> (accessed: 11.10.2016)
10. **Backx P., Wauters T., Dhoedt B., Demeester P.** A Comparison of Peer-to-Peer Architectures, *CiteSeerX [Official website]*, URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.9453&rep=rep1&type=pdf> (accessed: 10.11.2016).
11. **Freenet**, *Freenet [Official website]*, URL: <https://freenet-project.org> (accessed: 10.11.2016).
12. **Holz T., Steiner M., Dahl F., Biersack E., Freiling F.** Measurements and Mitigation of Peer-to-Peer based Botnets: A Case Study on Storm Worm, *Usenix [Official website]*, URL: https://www.usenix.org/legacy/events/leet08/tech/full_papers/holz/holz_html/index.html (accessed: 10.11.2016).
13. **Overnet**, *Wikipedia [Official website]*, URL: <https://en.wikipedia.org/wiki/Overnet> (accessed: 10.11.2016).
14. **Network Time Protocol**, *Wikipedia [Official website]*, URL: https://en.wikipedia.org/wiki/Network_Time_Protocol (accessed: 10.11.2016).
15. **Kademlia**, *Wikipedia [Official website]*, URL: <https://en.wikipedia.org/wiki/Kademlia> (accessed: 10.11.2016).
16. **Stover S., Dittrich D., Hernandez J., Dietrich S.** Analysis of the Storm and Nugache Trojans: P2P is Here, *Usenix [Official website]*, URL: <https://www.usenix.org/system/files/login/articles/526-stover.pdf> (accessed: 10.11.2016).
17. **Wang P., Sparks S., Zou C. C.** An Advanced Hybrid Peer-to-Peer Botnet, *Usenix [Official website]*, URL: https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/wang/wang_html/ (accessed: 10.11.2016).
18. **Starnberger G., Kruegel C., Kirda E.** Overbot — A Botnet Protocol Based on Kademlia, *Technische Universität Darmstadt [Official website]*, URL: https://www.p2p.tu-darmstadt.de/fileadmin/user_upload/Group_TK/overbot.pdf (accessed: 10.11.2016).
19. **Stock B., Gobel J., Engelberth M., Freiling F. C., Holz T.** Walowdac — Analysis of a Peer-to-Peer Botnet, *Academia Sinica [Official website]*, URL: http://mmnet.iis.sinica.edu.tw/botnet/file/20110117/20110117_1.pdf (accessed: 10.11.2016).
20. **Fast flux**, *Wikipedia [Official website]*, URL: https://en.wikipedia.org/wiki/Fast_flux (accessed: 10.11.2016).
21. **Lu T. T., Liao H. Y., Chen M. F.** An Advanced Hybrid P2P Botnet 2.0, *WASET [Official website]*, URL: <http://waset.org/publications/12157/an-advanced-hybrid-p2p-botnet-2.0> (accessed: 10.11.2016).
22. **Werner T.** Botnet Shutdown Success Story: How Kaspersky Lab Disabled the Hlux/Kelihos Botnet, *Securelist [Official website]*, URL: <https://securelist.com/blog/research/31058/botnet-shutdown-success-story-how-kaspersky-lab-disabled-the-hluxkelihos-botnet-15/> (accessed: 11.11.2016).
23. **Werner T.** The Miner Botnet: Bitcoin Mining Goes Peer-To-Peer, *Securelist [Official website]*, URL: <https://securelist.com/blog/incidents/30863/the-miner-botnet-bitcoin-mining-goes-peer-to-peer-33/> (accessed: 11.11.2016).
24. **Bitcoin**, *Wikipedia [Official website]*, URL: <https://en.wikipedia.org/wiki/Bitcoin> (accessed: 11.11.2016).
25. **Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus**, *SysSec [Official website]*, URL: http://www.syssec-project.eu/m/page-media/3/zeus_malware13.pdf (accessed: 11.11.2016).
26. **Domain generation algorithm**, *Wikipedia [Official website]*, URL: https://en.wikipedia.org/wiki/Domain_generation_algorithm (accessed: 11.11.2016).
27. **Sanatinia A., Noubir G.** OnionBots: Subverting Privacy Infrastructure for Cyber Attacks, *Arxiv [Official website]*, URL: <https://arxiv.org/pdf/1501.03378.pdf> (accessed: 11.11.2016).
28. **Tor**, *Wikipedia [Official website]*, URL: [https://en.wikipedia.org/wiki/Tor_\(anonymity_network\)](https://en.wikipedia.org/wiki/Tor_(anonymity_network)) (accessed: 11.11.2016).
29. **Avdoshin S. M., Lazarenko A. V.** Tekhnologiya anonimnyh setej [Technology of Anonymous Networks], *Informacionnye tehnologii*, 2016, vol. 22, no. 4, pp. 284—291 (in Russian).
30. **Avdoshin S. M., Lazarenko A. V.** Deep Web Users Deanonymization System. Proceedings of the Institute for System Programming, 2016, vol. 28, is. 3, pp. 21—34.
31. **Avdoshin S. M., Lazarenko A. V.** Metody deanonimizacii pol'zovatelej Tor [Tor Users Deanonymization Methods], *Informacionnye tehnologii*, 2016, vol. 22, no. 5, pp. 362—372 (in Russian).

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

УДК 004.932.72'1, 57.087.1

Ю. С. Ефимов, студент, e-mail: yuri.efimov@phystech.edu,
МФТИ, Московская область, г. Долгопрудный,

И. А. Матвеев, д-р техн. наук, зав. сектором, e-mail: matveev@ccas.ru,
ФИЦ ИУ РАН; Iritech Inc.

Выделение точных границ радужки на изображении глаза

Решается задача сегментации области радужки на изображении глаза человека. Область радужки в первоначальном приближении задается двумя неконцентрическими окружностями, которые определяются с использованием разновидности преобразования Хафа — метода парных градиентов. На этом шаге строится устойчивое решение, т. е. на большинстве изображений, содержащих радужку, метод определяет ее границы без грубых ошибок. Затем внутренняя граница уточняется методом оптимального кругового пути. Качество работы метода оценено на открытых базах изображений радужки.

Ключевые слова: выделение границ РОГ, выделение окружностей, методология Хафа, оптимальный круговой путь

Введение

Текстура радужной оболочки человеческого глаза является распространенным биометрическим признаком, используемым современными системами идентификации личности. Один из первых этапов применения любого алгоритма распознавания по данному признаку — выделение области радужки. Результат идентификации в значительной степени зависит от качества определения границ радужной оболочки. На изображении глаза радужка представляет собой множество пикселей, ограниченное внутри относительно темной областью — зрачком, а снаружи — относительно светлой областью — склерой. Зрачок большинства людей имеет форму, близкую к кругу. Внешняя часть радужной оболочки чаще имеет форму эллипса с малым эксцентриситетом [1] (порядка 6 %). В процессе выделения области радужки на изображении возникают трудности, связанные с шумами, бликами при съемке, а также возможным наличием посторонних объектов, таких как веки, закрывающие искомую область, линзы или части оправы очков. Примеры таких дефектов изображений приведены на рис. 1.



Рис. 1. Изображения глаз с дефектами

Одним из подходов к сегментации радужной оболочки является аппроксимация ее границ двумя окружностями. В одной из первых работ [2] в области выделения радужки на изображении глаза это осуществляется с помощью интегро-дифференциального оператора. Данный метод позволяет с достаточной точностью определять параметры границ и имеет высокую вычислительную сложность. Иногда делается предположение о том, что окружности внешней и внутренней границ радужки должны быть концентрическими [3]. Это не соответствует действительности: расстояние между их центрами может составлять до 10 % размеров радужки [4]. Зрачок на изображении глаза обычно представляет собой темную область округлой формы, поэтому для определения его параметров применяют бинаризацию [5, 6]. Для поиска зрачка на изображении глаза применяются подходы, основанные на математической морфологии, например, метод рекурсивной эрозии [6]. Морфологические методы в большинстве случаев позволяют грубо, но быстро оценить параметры зрачка, чтобы впоследствии уточнить их иным методом [7]. В работах [8—10] для выделения границ используется модель активных контуров.

Распространенным алгоритмом выделения объектов заданного класса на изображениях является преобразование Хафа. Оно находит на изображении кривые, заданные параметрически, в том числе и окружности. Идея преобразования Хафа заключается в поиске локальных максимумов в пространстве параметров. Для окружностей пространство параметров является трехмерным, что дает слож-

ность поиска максимумов $O(N^3)$. Применение преобразования Хафа к задаче поиска границ радужной оболочки с учетом их округлой формы впервые было исследовано Уильдсом в работе [11]. Существуют разновидности преобразования Хафа, позволяющие уменьшить вычислительную сложность. В работах [3, 6, 12], например, предлагаются подходы к детектированию окружностей, использующие информацию о градиенте яркости в каждой точке изображения для отбора точек интереса. В [3, 6] описывается метод выделения концентрических окружностей, использующий двухмерный массив-аккумулятор, в [12] предлагается использовать кривые равной освещенности и сложную систему голосования в пространстве параметров. Преобразование Хафа применяется и для уточнения параметров зрачка или радужки, поскольку процедура предварительной грубой оценки этих параметров сокращает перебор точек-кандидатов, принадлежащих гипотетическим окружностям на изображении [8].

Границы радужной оболочки приближаются окружностями с хорошей точностью, однако существуют задачи, в которых требуется более детальное описание границы. В особенности это касается границы зрачок—радужка. Как правило, зрачок человека близок по форме к кругу, но в большинстве случаев не является по форме идеально круглым, а имеет нерегулярные отклонения с относительной величиной 5...10 % (рис. 2). Таким образом, возникает задача определения на изображении контура округлой формы, окружающего темную область на более светлом фоне, с использованием уже построенной аппроксимации данного контура окружностью, иными словами, задача уточнения. В работе [13] для решения похожей задачи — выделения замкнутых границ — предложен метод оптимального кругового пути.

В данной работе предлагается модифицировать подход Хафа к детектированию окружностей с учетом специфики изображений радужки. Используется информация о направлениях перепадов яркости в точках границ радужной оболочки, разделенные массивы-аккумуляторы и новая модель голосования для поиска центра глаза. Полученная аппроксимация границ радужки окружностями уточняется попиксельно с применением алгоритма поиска оптимального кругового пути.

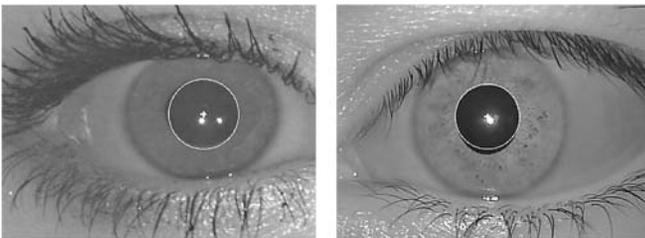


Рис. 2. Отклонения формы зрачка от округлой

Постановка задачи

Входные данные метода — черно-белое растровое изображение **I** размером $W \times H$. Каждый пиксель входного изображения кодируется одним байтом, т. е. может принимать целочисленные значения в интервале $[0; 255]$, что соответствует 256 градациям яркости. Требуется аппроксимировать внешнюю и внутреннюю границы радужной оболочки двумя окружностями. Полученное решение можно записать в виде

$$\omega = \{x_p, y_p, r_p, x_f, y_f, r_f\},$$

где x_p, y_p, r_p — координаты центра и радиус окружности, соответствующей границе зрачок—радужка, а x_f, y_f, r_f — соответственно параметры окружности, аппроксимирующей границу между радужкой и склерой.

Для оценки качества сегментации с помощью предлагаемого метода осуществляется сравнение результатов его работы с экспертной разметкой по набору изображений. Для входного изображения известны "истинные" параметры окружностей-границ радужной оболочки

$$\hat{\omega} = \{\hat{x}_p, \hat{y}_p, \hat{r}_p, \hat{x}_f, \hat{y}_f, \hat{r}_f\},$$

определенные человеком-экспертом вручную. Вводятся функционалы качества определения центров полученных окружностей

$$S_c(\omega) = |x_p - \hat{x}_p| + |y_p - \hat{y}_p| + |x_f - \hat{x}_f| + |y_f - \hat{y}_f|,$$

и качества определения радиусов

$$S_r(\omega) = |r_p - \hat{r}_p| + |r_f - \hat{r}_f|.$$

Также вводится суммарный функционал качества как сумма двух вышеописанных: $S(\omega) = S_c(\omega) + S_r(\omega)$.

Затем по значениям функционалов для данного изображения рассчитывается относительная ошибка определения параметров границ, как отношение значения соответствующего функционала к истинному радиусу внешней границы радужной оболочки:

$$\varepsilon_c(\omega) = \frac{S_c(\omega)}{\hat{r}_f}, \quad \varepsilon_r(\omega) = \frac{S_r(\omega)}{\hat{r}_f}, \quad \varepsilon(\omega) = \varepsilon_c(\omega) + \varepsilon_r(\omega).$$

Человек-эксперт может вручную разметить лишь небольшое число точных контуров зрачка на изображениях, поскольку такая разметка достаточно трудоемка и подвержена большому числу ошибок, нежели разметка приближенными окружностями. Поэтому полученный контур в виде последовательности координат $S^* = \{(x_k, y_k)\}_{k=1}^M$ можно заменить эквивалентным кругом $O^* = O(x^*, y^*, r^*)$, таким что его площадь и положение центра масс совпадают с центром масс области, окруженной S^* :

$$\begin{aligned} x^* &= \frac{1}{M} \sum_{k=1}^M x_k, \quad y^* = \frac{1}{M} \sum_{k=1}^M y_k, \\ r^* &= \frac{1}{M} \sum_{k=1}^M \sqrt{(x_k - x_p)^2 + (y_k - y_p)^2}, \end{aligned} \quad (1)$$

где M — число точек контура. Эти параметры также сравниваются с экспертной разметкой $\hat{\omega}$.

В итоге, качество решения задачи определяется с помощью распределения относительной ошибки для набора входных изображений.

Система методов выделения точных границ радужки

Выделение региона радужной оболочки глаза (РОГ) на изображении осуществляется в два этапа. На первом этапе с помощью модификации системы голосования для подхода Хафа определяются параметры окружностей, аппроксимирующих границы РОГ. В присутствии шумов, бликов от осветителя и искажений формы объектов на изображении подход Хафа может давать неточное решение данной задачи, поэтому на втором этапе осуществляется уточнение полученных аппроксимаций с учетом специфики границ зрачок—радужка и радужка—склера.

Метод парных градиентов

Введем обозначения:

- $\mathbf{q} = (x, y)$ — точка исходного растрового изображения;
- $\mathbf{g} = (g_x(x, y), g_y(x, y))$ — вектор градиента яркости в точке \mathbf{q} ;
- $\mathbf{p} = (x_c, y_c, r)$ — вектор параметров окружности;
- \mathbf{Q} — двумерное изображение-аккумулятор, соразмерное \mathbf{I} . Каждой точке аккумулятора (x, y) соответствуют центры некоторых гипотетических окружностей $\mathbf{p}: (x_c, y_c) = (x, y)$. Чем больше значение аккумулятора в точке (x^*, y^*) , тем вероятнее присутствие на исходном изображении некоторой окружности с центром $(x_c, y_c) = (x^*, y^*)$;
- $\mathbf{G} = \{x, y, g(x, y), \varphi\}$ — множество точек, принадлежащих границам объектов на изображении. Под границей объекта следует понимать множество точек с большим значением модуля градиента яркости. \mathbf{G} содержит координаты x и y гра-

ничной точки, величину $g(x, y)$ и угол направления φ градиента;

- $\mathbf{P} = \{\mathbf{q}_i, \mathbf{p}_i\}_{i=1}^m$ — множество пикселей-кандидатов вместе с соответствующими им параметрами гипотетических окружностей. Данные точки предположительно принадлежат границам радужки, а значит, их признаки представляют интерес для анализа.

Для поиска координат центра окружности используется изображение-аккумулятор $\mathbf{Q} \subseteq \mathbb{R}^2$, соразмерное исходному. Для поиска радиуса строится одномерный аккумулятор $H \subseteq \mathbb{R}$, размеры которого ограничены размерами \mathbf{I} . Для отбора пикселей-кандидатов используется информация о величине и направлении градиента яркости в точках \mathbf{I} .

Как известно, круг на плоскости можно задать вектором параметров: $\mathbf{p} = (x_c, y_c, r) \in \mathbb{R}^3$, где пара $(x_c, y_c) \in \mathbb{R}^2$ задает центр ограничивающей его окружности и $r \in \mathbb{R}$ определяет ее радиус. Предположим, что на изображении присутствует единственный круг с параметрами (x_c, y_c, r) . Тогда в точках его границы с окружающим фоном модуль градиента яркости $g(x, y)$ будет превосходить его значение в остальных точках изображения.

Рассмотрим пару точек $\mathbf{q}_1 = (x_1, y_1)$ и $\mathbf{q}_2 = (x_2, y_2)$, лежащих на границе круга и таких, что градиенты яркости $\mathbf{g}(\mathbf{q}_1)$ и $\mathbf{g}(\mathbf{q}_2)$ образуют между собой некоторый угол ψ (рис. 3). Каждая такая точка вместе с вектором градиента задает прямую

$$l_i = l(\mathbf{q}_i, \mathbf{g}(\mathbf{q}_i)) = \mathbf{q}_i - t_i \mathbf{g}(\mathbf{q}_i), t_i \in \mathbb{R}, i \in \{1, 2\}.$$

Тогда можно определить координаты точки пересечения $\hat{\mathbf{q}}$ данных прямых. И в нашем предположении для точек $\mathbf{q}_1, \mathbf{q}_2$, лежащих на окружности, ограничивающей данный круг, расстояния $\|\mathbf{q}_1 - \hat{\mathbf{q}}\|$ и $\|\mathbf{q}_2 - \hat{\mathbf{q}}\|$ будут равны.

Таким образом, можно сформулировать условия отбора точек интереса, принадлежащих границам гипотетических круглых областей и, соответственно, ограничивающих их окружностей на изображении:

$$\|\mathbf{g}(\mathbf{q}_1)\| > T_g; \quad (2)$$

$$\|\mathbf{g}(\mathbf{q}_2)\| > T_g; \quad (3)$$

$$\angle(\mathbf{g}(\mathbf{q}_1), \mathbf{g}(\mathbf{q}_2)) = \psi; \quad (4)$$

$$\|\mathbf{q}_1 - \hat{\mathbf{q}}\| = \|\mathbf{q}_2 - \hat{\mathbf{q}}\|, \quad (5)$$

где условия (2), (3) определяют отбор точек с градиентом яркости с нижним порогом T_g , (4) накладывает ограничение на угол между векторами градиентов, а (5) — на расстояние между парой точек и точкой пересечения прямых l_1 и l_2 .

Если пара точек $\{\mathbf{q}_1, \mathbf{q}_2\}$ удовлетворяет условиям отбора, то можно легко определить параметры соответствующей ей гипотетической окружности \hat{O} . Координаты ее центра (x_c, y_c) определяются координатами

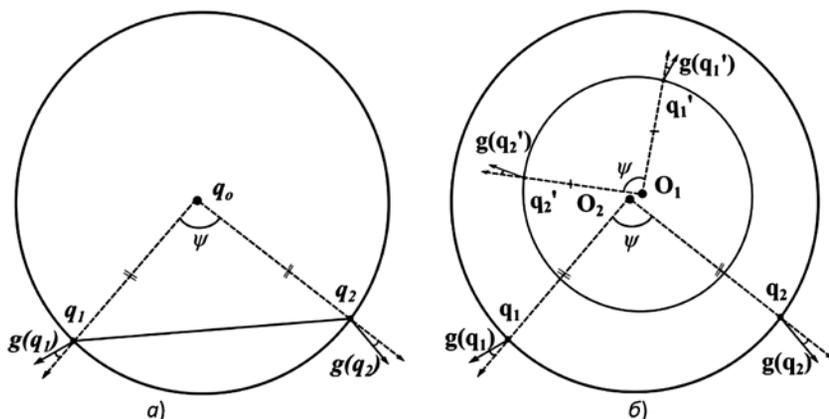


Рис. 3. Иллюстрация идеи метода парных градиентов: а — отбор граничных точек; б — случай двух окружностей

точки \mathbf{q} пересечения прямых $l_i = l(\mathbf{q}_i, \mathbf{g}(\mathbf{q}_i))$, а радиус находится с учетом (x_c, y_c) :

$$r = \|\mathbf{q}_1 - \hat{\mathbf{q}}\| = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}.$$

Присутствие на изображении глаза двух окружностей, соответствующих границам радужки, не накладывает ограничений на применение данного метода, поскольку условие (5) исключает возможность отбора пар точек с разных окружностей при выполнении условий (2)–(4) (рис. 3, б).

Метод оптимального кругового пути

В данной подзадаче известны относительно полные начальные данные — параметры окружности зрачка: координаты ее центра и радиус. Требуется найти контур границы зрачка как последовательность пикселей входного изображения. Поскольку искомый контур проходит вокруг центра глаза, имеет смысл провести полярное преобразование с полюсом в этой точке, что облегчает представление и дальнейшие вычисления.

Для изображения \mathbf{I} размерами $W \times H$ результат такого преобразования с полюсом в точке (x_p, y_p) имеет вид

$$\mathbf{I}_p(\rho, \varphi) = \begin{cases} \mathbf{I}(x_p + \lfloor \rho \cos(\varphi) \rfloor, y_p + \lfloor \rho \sin(\varphi) \rfloor), \\ \text{если } x_p + \lfloor \rho \cos(\varphi) \rfloor \in [1, W] \\ \text{и } y_p + \lfloor \rho \sin(\varphi) \rfloor \in [1, H]; \\ 0, \text{ иначе.} \end{cases}$$

Полярное преобразование переводит кольцо в прямоугольник из системы координат OXY в систему $O\rho\varphi$, где результат представляется матрицей интенсивностей \mathbf{I}_p размера $W_p \times H_p$ (рис. 4). Назовем данную матрицу полярным представлением. Исходная задача превращается в задачу нахождения оптимального пути между левой и правой сторонами прямоугольника при условии, что ординаты терминальных точек пути совпадают. Форма искомого контура близка к округлой, полюс преобразования находится в центре окружности, аппроксимирующей контур, поэтому полярное представление контура однозначно относительно угла, т. е. каждому значению угла соответствует одно и только одно значение радиуса контура. Тогда искомый контур параметризуется функцией $\rho(\varphi)$. Он не проходит вблизи полюса преобразования, значит, производная значения радиуса по углу ограничена:

$$\frac{\partial \rho}{\partial \varphi} < C_1. \quad (6)$$

Можно выбрать параметры полярного преобразования так, чтобы $C_1 \leq 1$.

На полярном представлении \mathbf{I}_p искомый контур S_p представляется как последовательность пикселей:

$$S_p = \{\rho(\varphi_k)\}_{k=1}^M.$$

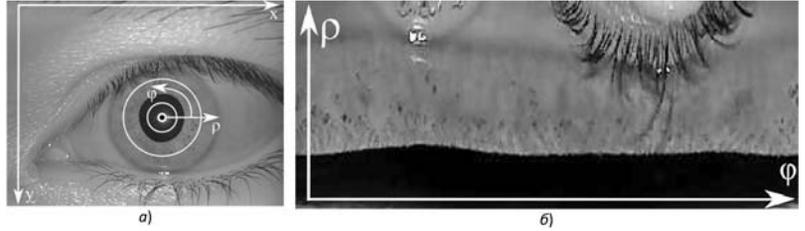


Рис. 4. Пример полярного преобразования: а — область границы зрачка; б — ее полярное представление

Подразумевается, что в каждом столбце \mathbf{I}_p содержится ровно один ее элемент. В силу (6) должно выполняться условие $\forall k \in [1, M-1] \Rightarrow |\rho(\varphi_{k+1}) - \rho(\varphi_k)| \leq 1$, а замкнутость искомого контура в декартовых координатах требует $|\rho(\varphi_1) - \rho(\varphi_M)| \leq 1$.

Тогда при обходе контура по пикселям шаг по угловой координате будет единичным, т. е. $\varphi_k = k$ и $M = W_p$, а сам контур может быть представлен последовательностью значений радиуса: $S_p = \{\rho_k\}_{k=1}^{W_p}$.

Рис. 5 показывает возможные пути между соседними точками контура, если проследживать его слева направо. Из точки A с координатами $(\varphi, \rho) = (2, 1)$ можно проследовать в точки $A_1 = (3, 1)$ и $A_2 = (3, 2)$, из точки $B = (5, 2)$ можно проследовать в точки $B_1 = (6, 1)$, $B_2 = (6, 2)$ и $B_3 = (6, 3)$.

Для нахождения оптимального контура задается некоторая числовая характеристика для любого S_p в зависимости от его формы и местоположения на \mathbf{I}_p . Введем функцию "стоимости" перехода между точками изображения из столбцов (n, ρ_n) и (m, ρ_m) : $C(\rho_n, \rho_m) = C_0(\rho_n, \rho_m) + C_1(\rho_n, \rho_m)$. Первое слагаемое $C_0(\rho_n, \rho_m)$ определяется локальными свойствами самого полярного представления:

$$C_0(\rho_n, \rho_m) = w(n, \rho_n), \quad (7)$$

где $w(n, \rho_n)$ — функция интенсивностей окрестности точки контура, о которой будет рассказано ниже. Вклад второго слагаемого $C_1(\rho_n, \rho_m)$ определяется лишь формой искомого контура и должен быть минимален для прямых горизонтальных линий на изображении, которые в исходной системе координат OXY соответствуют окружностям:

$$C_1(\rho_n, \rho_m) = \begin{cases} 0, & \text{если } \rho_n = \rho_m; \\ T_1, & \text{если } |\rho_n - \rho_m| \leq 1; \\ \infty, & \text{иначе.} \end{cases}$$

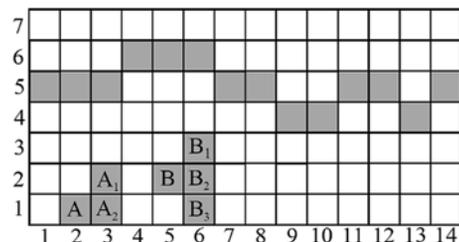


Рис. 5. Возможные переходы между точками контура

Константа T_1 определяет рост стоимости прохода по данному контуру при отклонении его формы от округлой и зависит лишь от параметров полярного преобразования.

Для заданного пути $S_p = \{\rho_k\}_{k=1}^{W_p}$ рассчитывается общая стоимость:

$$C(S_p) = \sum_{k=1}^{W_p} C(\rho_k, \rho_{k+1}). \quad (8)$$

Оптимальный контур — это контур с минимальной общей стоимостью:

$$S_p^* = \underset{(x, y)}{\operatorname{argmin}} C(S_p). \quad (9)$$

Применение методов к сегментации радужки

Рассмотрим процесс точного выделения границ радужной оболочки с помощью данных методов более подробно.

Шаг 1. Отбор граничных точек.

Исходное изображение представляет собой заданную на прямоугольной сетке размером $W \times H$ дискретную функцию интенсивностей $I(x, y)$. Для отбора граничных точек применяется фильтр Кэнни [14], использующий в своей работе поле градиентов яркости. Результат применения данного фильтра определяется набором входных параметров: значениями нижнего и верхнего порогов для градиента яркости $t_{low}, t_{high} \in [0; 1]$ соответственно и размером σ фильтра Гаусса, используемого для сглаживания входного изображения в целях подавления шумов. Значения порогов позволяют отбросить фрагменты слишком слабых границ и снизить число разрывов в выделяемых фрагментах. Для расчета градиентов яркости используются маски Собеля, например

$$S_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, S_y = S_x^T = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \quad (10)$$

и поля градиентов яркости рассчитываются как дискретная свертка соответствующих матриц с функцией яркости изображения:

$$g_x(x, y) = S_x * I(x, y), g_y(x, y) = S_y * I(x, y).$$

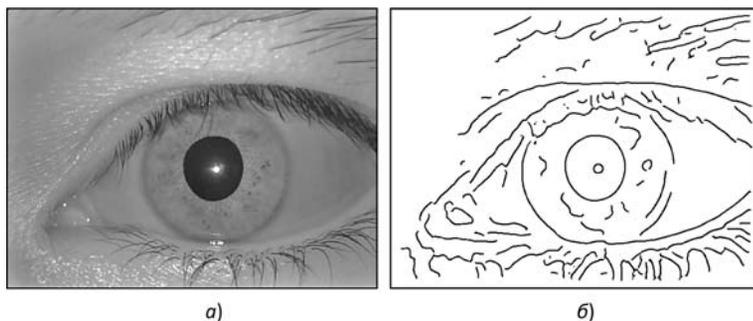


Рис. 6. Пример построения множества граничных точек: а — исходное изображение; б — множество граничных точек

С помощью масок Собеля (10) строится изображение \mathbf{V} , отражающее распределение величин градиентов яркости $g(x, y)$.

Для величин t_{low}, t_{high} принимаются следующие значения:

$$t_{low} = \max\{0,55, 0,7t\}, t_{high} = \min\{0,9, 1,3t\},$$

где t — порог Оцу для данного изображения, нормированный на максимальное значение яркости.

Формируется множество точек предполагаемых границ объектов на изображении $\mathbf{G} = \{x, y, g(x, y), \varphi(x, y)\}$. Данное множество можно наглядно представить в виде бинарного изображения \mathbf{B} (рис. 6, б), соразмерного исходному (рис. 6, а):

$$\mathbf{B}(x, y) = \begin{cases} 1, & \text{если } (x, y) \in \mathbf{G}; \\ 0, & \text{иначе.} \end{cases}$$

Шаг 2. Поиск наиболее выраженной границы.

Множество \mathbf{G} используется для определения параметров всевозможных гипотетических окружностей на изображении с помощью метода парных градиентов. Среди точек $\mathbf{q} \in \mathbf{G}$ отбираются все пары $\{\mathbf{q}_1, \mathbf{q}_2\}$, удовлетворяющие условиям (4), (5), т. е. имеющие угол ψ между направлениями соответствующих градиентов.

Пусть имеется N граничных точек, тогда сложность перебора всевозможных их пар равна $O(N^2)$. Для уменьшения вычислительной сложности используется метод перебора "скользящим окном". С учетом природы входных данных делается предположение, что всевозможные направления градиентов яркости на изображении распределены приблизительно равномерно на отрезке $[-\pi, \pi]$. Тогда в отсортированном по величине угла φ массиве граничных точек поиск пар можно осуществить за $O(N)$. Первым в отсортированном массиве будет элемент $\hat{\mathbf{q}}_0$ со значением $\varphi = \varphi_0 = -\pi$. "Скользящее окно" шириной 10° определяется вокруг первого элемента с углом $\varphi_j > \varphi_0 + \psi$. Внутри этого "окна" в данном предположении содержатся возможные пары для $\hat{\mathbf{q}}_0$. Данным способом осуществляется однократный проход массива.

Для каждой отобранной пары точек определяются параметры соответствующей предполагаемой окружности. При отборе пар согласно условиям (4), (5) используется порог $T_\phi = \arccos(10^\circ) =$

$$= 0,984. \text{ Формируется множество параметров}$$

интереса $\mathbf{P} = \{\mathbf{q}_i, \mathbf{p}_i\}_{i=1}^M$, которое используется

при голосовании методом Хафа в массиве-аккумуляторе \mathbf{Q} для поиска центра окружности наиболее выраженной границы радужки. Максимум $\mathbf{q}_1^* = (x_1^*, y_1^*) = \underset{(x, y)}{\operatorname{argmax}} \mathbf{Q}(x, y)$ соответствует такому центру (рис. 7).

Найденная позиция \mathbf{q}_1^* позволяет исключить шумовые элементы из \mathbf{G} . Для граничных точек \mathbf{q} , принадлежащих контуру окружности

с данным центром \mathbf{q}_1^* , угол между направлением $\mathbf{g}(\mathbf{q})$ и вектором $\mathbf{d} = \mathbf{q} - \mathbf{q}_1^*$ должен быть мал (рис. 8):

$$\arccos \frac{\mathbf{d} \cdot \mathbf{g}(\mathbf{q})}{\|\mathbf{d}\| \|\mathbf{g}(\mathbf{q})\|} < 10^\circ. \quad (11)$$

Для определения радиуса строится нормированная гистограмма $H(r)$ расстояний от \mathbf{q}_1^* до точек \mathbf{G} :

$$H(r) = \frac{1}{r} |\{\mathbf{q} \in \mathbf{G} : \|\mathbf{q} - \mathbf{q}_1^*\| \in (r - 0,5; r + 0,5)\}|.$$

Для устранения шума применяется усреднение с помощью одномерного приближения гауссиана

$\frac{x^2}{2\sigma^2}$ шириной $\sigma = 10,0$. Максимум результирующей гистограммы $H(r)$ соответствует искомому радиусу r_1^* (рис. 9, а).

Шаг 3. Поиск второй границы радужки.

Для поиска второй границы используются предельные соотношения между радиусами радужной оболочки и зрачка, полученные на основании статистических данных [6]:

$$r_p > \frac{1}{7} r_f; \quad (12)$$

$$r_l < \frac{3}{4} r_p; \quad (13)$$

$$r_p > \sqrt{(x_l - x_p)^2 + (y_l - y_p)^2}, \quad (14)$$

где (x_p, y_p, r_p) — параметры внешней границы радужки, а (x_p, y_p, r_p) — аналогичные параметры зрачка. Неравенство (12) означает, что радиус радужной оболочки не может превосходить радиус зрачка более, чем в 7 раз. Неравенство (13) вводит ограничение с другой стороны: радиус зрачка не может достигать 75 % радиуса радужки. Неравенство (14) утверждает, что центр радужки лежит внутри зрачка.

Для учета наличия неконцентрических границ из \mathbf{P} исключаются все точки, соответствующие гипотетическим окружностям с центрами вне найденной, а также с $r \in [0; \frac{1}{7}r_1^*] \cup [\frac{3}{4}r_1^*; \frac{4}{3}r_1^*]$, и процедура парного голосования повторяется снова. Для определенного центра $\mathbf{q}_2^* = (x_2^*, y_2^*) = \underset{(x, y)}{\operatorname{argmax}} \mathbf{Q}(x, y)$ строится новая гистограмма расстояний до граничных точек $\hat{H}(r)$, значения которой для столбцов $r \in [0; \frac{1}{7}r_1^*] \cup [\frac{3}{4}r_1^*; \frac{4}{3}r_1^*]$ обнуляются в соответствии с условиями (12)—(14), чтобы исключить возможность повторного детектирования уже найденной границы. Искомый радиус r_2^* определяется аналогично шагу 2 (рис. 9, б).

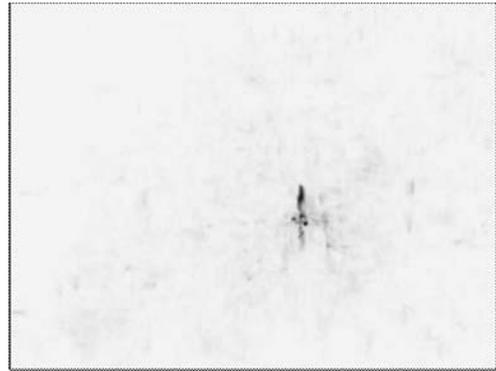


Рис. 7. Аккумулятор для поиска центра

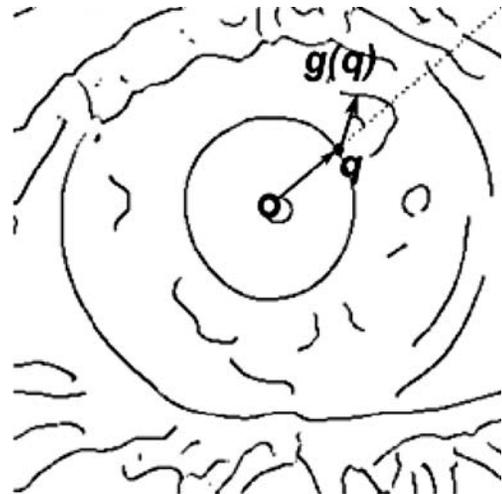


Рис. 8. Карта точек, удовлетворяющих условию (11)

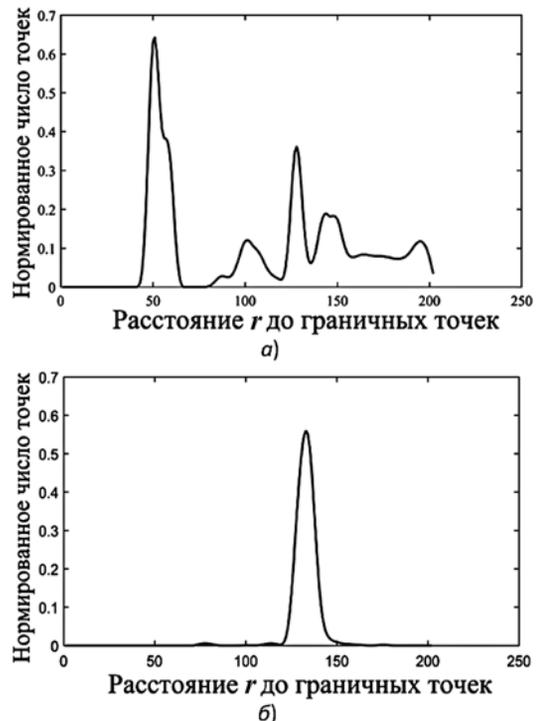


Рис. 9. Вид гистограммы для поиска радиусов (а) и результат обнуления ее столбцов (б)

Шаг 4. Определение точной границы зрачка.

Рассмотрим полярное представление \mathbf{I}_p круговой полосы \mathbf{I}_c границы зрачка:

$$\mathbf{I}_c = \{(x, y) \in \mathbf{I}: (x - x_p)^2 + (y - y_p)^2 = r^2, r \in [r_p - 20; r_p + 20]\}.$$

Для решения описанной задачи дискретной оптимизации (9) существуют жадные алгоритмы [13], однако, при небольшой высоте полярного представления $H_p < 40$ вычислительно проще полный перебор по столбцам \mathbf{I}_p , т. е. по точкам с одним значением φ . Обозначим стоимость перехода из точки $(0, \rho_0)$ в точку (m, ρ_m) как $C(\rho_0, \rho_m) = C_{(m)}(\rho_0, \rho_m)$. Поскольку ρ_0 и ρ_m меняются в пределах $[1; H_p]$, необходимо вычислить H_p^2 значений стоимости $C_{(m)}(\rho_0, \rho_m)$. Цена достижения точки в столбце $m + 1$ есть сумма цены достижения точки ρ_m в предыдущем столбце и цены перехода между точками соседних столбцов:

$$\begin{aligned} & C_{(m+1)}(\rho_0, \rho_{m+1}) = \\ & = \min_{\rho_{m+1}} \{C_{(m)}(\rho_0, \rho_m) + C(\rho_m, \rho_{m+1})\} = \\ & = \min \left\{ \begin{array}{l} C_{(m)}(\rho_0, \rho_m) + w(m, \rho_m); \\ C_{(m)}(\rho_0, \rho_m) + w(m, \rho_{m+1}) + T_1; \\ C_{(m)}(\rho_0, \rho_m) + w(m, \rho_{m-1}) + T_1. \end{array} \right. \end{aligned}$$

Путь, который дал минимум стоимости, запоминается. На последнем шаге получаются H_p^2 значений $C_{(W_p+1)}(\rho_0, \rho_{W_p+1})$. При этом только значения стоимостей с $|\rho_0 - \rho_{W_p+1}| \leq 1$ соответствуют замкнутым контурам и в задаче минимизации рассматриваются только они. Определяется последняя точка оптимального контура $\rho_{W_p+1}^* = \operatorname{argmin}_{\rho_{W_p+1}} C_{(W_p+1)}(\rho_0, \rho_{W_p+1})$.

Из известной последней точки легко восстанавливается сам оптимальный контур.

Рассмотрим первое слагаемое $C_0(\varphi, \rho_n) = w(\varphi, \rho_n)$, входящее в стоимость прохода через точку (φ, ρ_n) . Из постановки задачи уточнения ясно, что функция $w(\varphi, \rho)$ должна быть построена так, чтобы быть малой в точках, соответствующих контуру, и большой — в других. Используется модуль градиента

яркости. В точках контура он велик, поэтому точки с малым значением этого параметра должны вносить больший вклад в функцию стоимости. Также используется ограничение (11) на направление градиентов яркости в точках предполагаемого контура, который должен иметь круглую форму.

На изображении V величин градиентов яркости для точек кольца

$$V_c = \{(x, y): (x - x_p)^2 + (y - y_p)^2 = r^2, r \in [r_p - 20; r_p + 20]\}$$

проверяется выполнение условия (11). Строится полярное представление кольца V_c с полюсом в точке (x_p, y_p) (рис. 10).

В результате получается изображение \mathbf{V}_p размерами $W_p \times H_p$ области границы зрачка на изображении, $W_p = 360$, $H_p = 40$. Для обращения к точкам \mathbf{V}_p теперь используются координаты (n, ρ) : $n \in [1; 360]$, $\rho \in [1; 40]$.

Значения \mathbf{V}_p инвертируются и используются при построении функции $w(n, \rho)$ из формулы (7): $w(n, \rho) = \mathbf{V}_p(n, \rho)$. Таким образом, все слагаемые, составляющие функцию стоимости, определены, что позволяет решить задачу поиска пути минимальной стоимости $S_p^* = \{\rho_k\}_{k=1}^{W_p}$ методом перебора. С помощью обратного полярного преобразования из S_p^* формируется список координат точек

$$\text{границы зрачка } S^* = \{(x_k, y_k)\}_{k=1}^{W_p}.$$

Вычислительный эксперимент

Целью вычислительного эксперимента является проверка качества работы алгоритма на реальных изображениях глаз. При тестировании использовались растры разрешением 640×480 точек из баз изображений радужки CASIA [15] в количестве 2400 шт., ND-IRIS [16] в количестве 20 000 шт. и UBI [17] в количестве 1207 шт. Для каждого входного изображения экспертом были определены параметры зрачка и радужной оболочки. В эксперименте использовался персональный компьютер с процессором Intel Core i5-2450M с частотой 2,5 GHz.

Метод парных градиентов

Данный метод для отбора пикселей-кандидатов при голосовании использует параметр ψ — угол между парами градиентов яркости. Был проведен эксперимент, определяющий оптимальное значение для данного параметра с точки зрения точности решения задачи сегментации радужки. Метод парных градиентов с разными значениями $\psi \in [10^\circ; 180^\circ]$ был применен к 2000 изображений из указанных

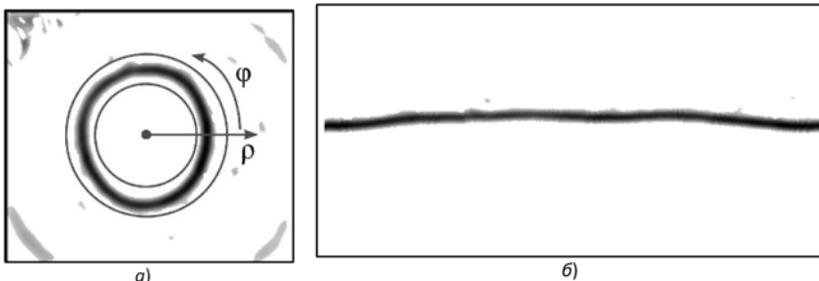


Рис. 10. Полярное представление (б) карты величин градиентов области границы зрачка (а)

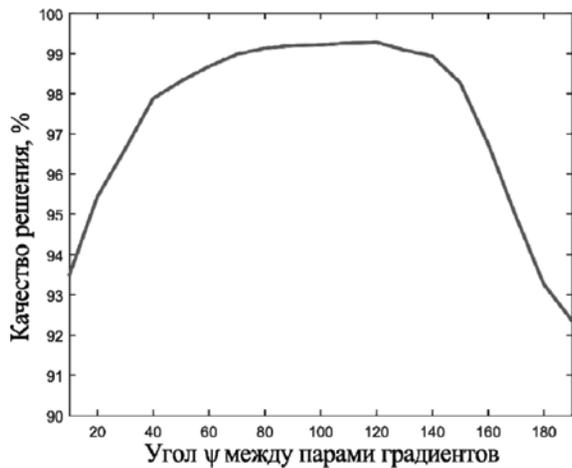


Рис. 11. Качество решения в зависимости от угла ψ

Таблица 1

Суммарная относительная ошибка выделения зрачка

Диапазон ошибки	CASIA		ND-IRIS		UBI	
	МПГ	ОКП	МПГ	ОКП	МПГ	ОКП
$\varepsilon \leq 0,1$	2226	2261	19 038	19 611	768	1016
$0,1 \leq \varepsilon \leq 0,2$	121	110	759	259	230	39
$0,2 \leq \varepsilon \leq 0,3$	24	13	67	37	6	5
$0,3 \leq \varepsilon \leq 0,4$	1	6	48	36	30	11
$0,4 \leq \varepsilon \leq 0,5$	4	1	41	19	42	19
$0,5 \leq \varepsilon \leq 0,6$	2	1	15	8	31	21
$0,6 \leq \varepsilon \leq 0,7$	1	3	8	4	15	11
$0,7 \leq \varepsilon \leq 0,8$	3	1	4	5	3	6
$0,8 \leq \varepsilon \leq 0,9$	0	0	0	1	5	3
$0,9 \leq \varepsilon \leq 1$	0	0	0	0	4	3
$\varepsilon \geq 1$	4	4	20	20	73	73
Всего	2400	2400	20 000	20 000	1207	1207

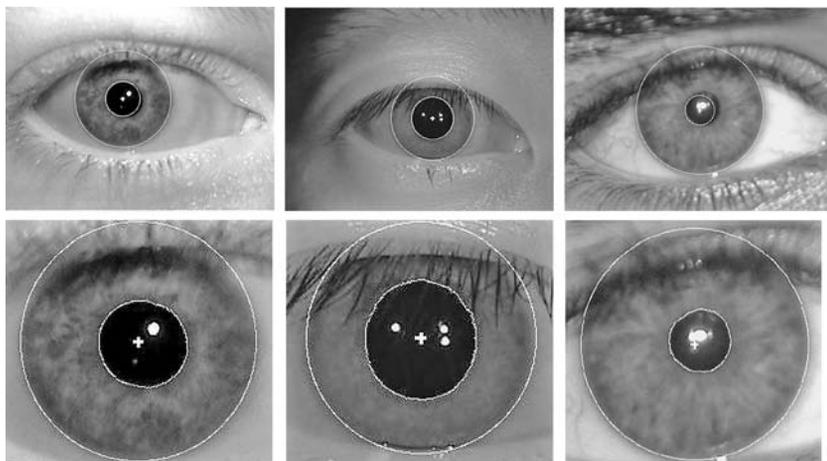


Рис. 12. Примеры корректной сегментации изображения глаза

баз, а затем с помощью экспертной разметки было оценено качество решения, которое определялось долей изображений, для которых значение относительной ошибки определения центра глаза не превосходит 20 %.

Результаты представлены на рис. 11. Как видно из графика, наилучшее качество решения достигается при $\psi = 120^\circ$. Дальнейший анализ метода парных градиентов проводится при данном значении угла ψ . Для каждого входного изображения по данным, размеченным экспертом, были рассчитаны значения абсолютной и относительной ошибки. Результаты эксперимента представлены в табл. 1 в колонках "Метод парных градиентов" (МПГ).

Метод оптимального кругового пути

По полученным в результате применения данного метода контурам зрачков $S^* = \{(x_k, y_k)\}_{k=1}^W$ строились эквивалентные круги (1). Результаты сравнения их параметров с экспертной разметкой приведены в табл. 1 в колонке "Оптимальный круговой путь" (ОКП).

Примеры сегментации радужной оболочки методом парных градиентов с последующим уточнением границы зрачка приведены на рис. 12.

Сравнение с существующими аналогами

В табл. 2 сопоставлены время вычисления и ошибки определения параметров зрачка для результата суммарного применения данных методов сегментации радужной оболочки. Сравнение проводилось с методами, описанными Wildes [11], Daugman [2, 10], Masek [18], Ma et al. [19].

Следует отметить, что для предлагаемого в работе метода описано полное время исполнения, в то время как для сравниваемых — только время определения зрачка. Применение метода парных градиентов с последующим уточнением границы зрачка методом оптимального кругового пути уступает по качеству сегментации и скорости исполнения методу Даугмана [10]. Метод, описанный в [18], также выигрывает по скорости, но проигрывает по точности сегментации зрачка практически в два раза, как и другие сравниваемые методы.

Таблица 2

Сравнение качества сегментации, определяемое различными методами

База изображений	Средняя относительная ошибка, %					
	Daugman	Daugman New	Ma et al.	Wildes	Masek	МПГ + ОКП
CASIA	2,15	1,19	4,79	5,37	5,15	2,14
ND-IRIS	2,13	1,79	5,92	6,33	5,59	1,57

Заключение

Предложен алгоритм выделения области радужки на изображении, позволяющий аппроксимировать ее границы окружностями, а контур зрачка определить как последовательность пик-

селей. Для оценки работоспособности алгоритма проведен вычислительный эксперимент на изображениях глаз разного качества. Осуществлено сравнение качества сегментации радужной оболочки с существующими в современной биометрии аналогами. Результаты представлены в таблицах.

Предлагаемая в данной работе модификация преобразования Хафа позволяет существенно сократить перебор граничных точек при определении центров окружностей. Контур, полученный в результате применения метода поиска оптимального кругового пути, можно использовать для уточнения круговой аппроксимации границы зрачка.

Список литературы

1. **Kansky J. J.** Clinical Ophthalmology: a Systematic Approach. London: Elsevier, 2003.
2. **Daugman J.** How iris recognition works // Proc. Int. Conf. Image Processing. 2002. Vol. 1. P. 33–36.
3. **Cauchie J., Fiolet V., Villers D.** Optimization of an Hough transform algorithm for the search of a center // Pattern Recognition. 2009. Vol. 41, N. 2. P. 567–574.
4. **Ibrahim M. T., Khan T. M., Khan A.** et al. A Novel and Efficient Feedback Method for Pupil and Iris Localization // Lecture Notes in Computer Science. 2011. Vol. 97, N. 1. P. 1066–1069.
5. **Koh J., Govindajaru V., Chaundhary V.** A Robust Iris Localization Method Using an Active Contour Model and Hough Transform // Proc. 20th Int. Conf. Pattern Recognition. 2010. P. 2852–2856.
6. **Ганькин К. А., Гнеушев А. Н., Матвеев И. А.** Сегментация изображения радужки, основанная на приближенных методах с последующими уточнениями // Известия РАН. Теория и системы управления. 2014. № 2. С. 80–94.

7. **Fernandez C., Perez D., Segura C.** et al. A Novel Method for Low-constrained Iris Boundary Localization // Proc. 5th IAPR Int. Conf. Biometrics. 2012. P. 291–296.
8. **Trucco E., Razeto M.** Robust iris localization in close-up images of the eye // Pattern Analysis and Applications. 2005. Vol. 8, N. 3. P. 247–255.
9. **Ross A., Shah S.** Segmenting non-ideal irises using geodesic active contours // Biometrics Symp.: Special Session on Research at the Biometric Consortium Conf. Baltimore. 2009. Vol. 4, N. 4. P. 824–836.
10. **Daugman J.** New Methods in Iris Recognition // IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics. 2007. Vol. 37, N. 5. P. 1167–1175.
11. **Wildes R. P.** Iris recognition: an emerging biometric technology // Proceedings of the IEEE. 1997. Vol. 85, N. 9. С. 1348–1363.
12. **Valenti R., Gevers T.** Eye Center Location through Invariant Isocentric Patterns // IEEE Transactions on Pattern Analysis and Machine Intelligence archive. 2012. Vol. 34, N. 9. P. 1785–1798.
13. **Sun C., Pallottino S.** Circular shortest path in images // Pattern Recognition. 2003. Vol. 36, N. 3. P. 709–719.
14. **Canny J. F.** A Computational Approach to Edge Detection // Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. 8, N. 6. P. 679–698.
15. **Chinese Academy of Science: Institute of Automation Iris Image Database.** Version 2.0. URL: <http://biometrics.idealtest.org/downloadDB.do>
16. **Phillips P. J., Scruggs W. T., O'Toole A. J.** Frvt2006 and ice2006 large-scale experimental results // IEEE Pattern Analysis and Machine Intelligence. 2010. Vol. 32, N. 5. P. 831–846.
17. **Proenca H., Alexandre H.** UBIRIS: A noisy iris image database // 13th International Conference on Image Analysis and Processing. 2005. P. 970–977.
18. **Masek L.** Recognition of human iris patterns for biometric. 2003. URL: <http://www.csse.uwa.edu.au>
19. **Ma L., Tan T., Wang Y. and Zhang D.** Efficient Iris Recognition by Characterizing Key Local Variations // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Vol. 13, N. 6. P. 739–750.

Yu. S. Efimov, Student, e-mail: yuri.efimov@phystech.edu,

Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region,

I. A. Matveev, Sci. D., Head of sector, e-mail: matveev@ccas.edu,

Federal Research Centre "Computing Centre" of Russian Academy of Sciences, Moscow, Russia

Iris Image Segmentation by Paired Gradient Method with Pupil Boundary Refinement

Locating the iris area in an eye image is an important part of any iris recognition system. In many studies iris boundaries are modelled as circles and Hough transform is used to obtain their parameters precisely. The use of mentioned object detection algorithm is limited in case of noisy images or the occlusion of the iris by eyelashes, glasses, hair etc. Original Hough transform for circles also has high computational complexity. Many modifications were proposed to improve the robustness and speed of the method. But the circular iris boundary representation is insufficient in terms of segmentation accuracy, mainly because of human pupil shape variations. Therefore, more precise representation is required.

The proposed method consists of two main steps. The first step is rough iris center and boundary radii search using the modification of Hough methodology, named Gradient Pair method. Image is processed with Canny filter and pairs of pixels are selected from the resulting boundaries, which most likely belong to one circle-like iris area border. The second step is the pupil border refinement using the circular shortest path method. The retrieved circular path is also used to improve the quality of eye center search.

Computational experiment is performed on data from the public iris image databases to obtain the optimal parameters for Gradient Pair method and check the overall efficiency of the algorithm.

The proposed method provides the high quality of eye center and pupil boundary search. The iris segmentation accuracy is comparable to that of modern state-of-art methods and outperforms many of them.

Keywords: iris image segmentation, circle detection, Hough transform, circular shortest path

References

1. **Kansky J. J.** *Clinical Ophthalmology: a Systematic Approach*. Elsevier, London, 2003.
2. **Daugman J.** How iris recognition works, *Proc. Int. Conf. Image Processing*, 2002, vol. 1, pp. 33–36.
3. **Cauchie J., Fiolet V., Villers D.** Optimization of an Hough transform algorithm for the search of a center, *Pattern Recognition*, 2009, vol. 41, no. 2, pp. 567–574.
4. **Ibrahim M. T., Khan T. M., Khan A.** et al. A Novel and Efficient Feedback Method for Pupil and Iris Localization, *Lecture Notes in Computer Science*, 2011, vol. 97, no. 1, pp. 1066–1069.
5. **Koh J., Govindajaru V., Chaundhary V.** A Robust Iris Localization Method Using an Active Contour Model and Hough Transform, *Proc. 20th Int. Conf. Pattern Recognition*, 2010, pp. 2852–2856.
6. **Gankin K. A., Gneushev A. N. and Matveev I. A.** Iris image segmentation based on approximate methods with subsequent refinements, *Journal of Computer and Systems Sciences International*. 2014, vol. 53, no. 2, pp. 80–94.
7. **Fernandez C., Perez D., Segura C. et al.** A Novel Method for Low-constrained Iris Boundary Localization, *Proc. 5th IAPR Int. Conf. Biometrics*, 2012, pp. 291–296.
8. **Trucco E., Razeto M.** Robust iris localization in close-up images of the eye, *Pattern Analysis and Applications*, 2005, vol. 8, no. 3, pp. 247–255.
9. **Ross A., Shah S.** Segmenting non-ideal irises using geodesic active contours, *Biometrics Symp.: Special Session on Research at the Biometric Consortium Conf. Baltimore*, 2009, vol. 4, no. 4, pp. 824–836.
10. **Daugman J.** New Methods in Iris Recognition, *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 2007, vol. 37, no. 5, pp. 1167–1175.
11. **Wildes R. P.** Iris recognition: an emerging biometric technology, *Proceedings of the IEEE*, 1997, vol. 85, no. 9, pp. 1348–1363.
12. **Valenti R., Gevers T.** Eye Center Location through Invariant Isocentric Patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*, 2012, vol. 34, no. 9, pp. 1785–1798.
13. **Sun C., Pallottino S.** Circular shortest path in images, *Pattern Recognition*, 2003, vol. 36, no. 3, pp. 709–719.
14. **Canny J. F.** A Computational Approach to Edge Detection, *Transactions on Pattern Analysis and Machine Intelligence*, 1986, vol. 8, no. 6, pp. 679–698.
15. **Chinese Academy of Science: Institute of Automation Iris Image Database**, Version 2.0. URL: <http://biometrics.idealtest.org/downloadDB.do>
16. **Phillips P. J., Scruggs W. T., O'Toole A. J.** Frvt2006 and ice2006 large-scale experimental results, *IEEE Pattern Analysis and Machine Intelligence*, 2010, vol. 32, no. 5, pp. 831–846.
17. **Proenca H., Alexandre H.** UBIRIS: A noisy iris image database, *13th International Conference on Image Analysis and Processing*, 2005, pp. 970–977.
18. **Masek L.** *Recognition of human iris patterns for biometric*. 2003, URL: <http://www.csse.uwa.edu.au>
19. **Ma L., Tan T., Wang Y. and Zhang D.** Efficient Iris Recognition by Characterizing Key Local Variations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 13, no. 6, pp. 739–750.

УДК 004.67

А. В. Голов^{1, 2}, аспирант, e-mail: golov.andrey@hotmail.com,

Е. А. Тимме^{2, 3}, канд. техн.наук, науч. сотр., e-mail: alpdem@yandex.ru,

А. В. Козлов^{2, 4}, гл. специалист, аспирант, e-mail: 89165363085a@mail.ru

¹Московский физико-технический институт (государственный университет), Москва

²Государственное казенное учреждение "Центр инновационных спортивных технологий и подготовки сборных команд" Москомспорта, Москва

³Государственный научный центр Институт медико-биологических проблем Российской академии наук, Москва

⁴Российский государственный университет физической культуры, спорта, молодежи и туризма, Москва

Алгоритм автоматизированной оценки параметров работоспособности человека при выполнении нагрузочных тестов

Разработан алгоритм автоматизированной оценки аэробного и анаэробного порогов по результатам нагрузочного тестирования с возрастающей нагрузкой на основе кусочно-линейной робастной регрессии. Алгоритм реализован в виде программного модуля, который может быть использован в задачах спортивной и авиационно-космической медицины.

Ключевые слова: функциональные нагрузочные тесты, аэробный порог, порог анаэробного обмена, эргометры, протокол нагрузочного тестирования, уровень работоспособности, робастная кусочно-линейная регрессия, метод бутстреппинга

Введение

Нагрузочные физиологические тесты являются универсальным средством определения индивидуальных параметров работоспособности человека, характеризующих режимы энергообеспечения мышечной деятельности. По этой причине примене-

ние таких тестов имеет широкое распространение в спортивной и авиационно-космической медицине. Эти тесты различаются по своему целевому назначению, применяемым эргометрам и используемым протоколам тестирования [1]. Основной целью тестирования является определение физической работоспособности и оценка ее важнейших пара-

метров: аэробного порога (АЭП) и порога анаэробного обмена (ПАНО) [2].

Во время тестирования регистрируются показатели, получаемые неинвазивными методами (потребление кислорода, выделение углекислого газа, легочная вентиляция, частота сердечных сокращений), а также показатель, получаемый инвазивным методом — концентрация лактата в капиллярной крови. Наиболее полную информацию дают тесты с повышающейся нагрузкой по линейному и ступенчатому протоколам, в которых практически достигается уровень максимального потребления кислорода и максимальное утомление [3].

На результаты тестирования могут влиять физиологическое и психологическое состояния человека, предшествующие физические нагрузки, время и состав принимаемой пищи в день теста, характер предварительной разминки [4]. В связи с этим спортсмен получает инструкции, соблюдение которых стандартизирует условия проведения тестирования по всем перечисленным пунктам.

Напряжение всех физиологических систем спортсмена, обеспечивающих мышечную работу, формирует кислородный запрос, удовлетворяемый работой дыхательной и сердечно-сосудистой систем, показатели которых, при сопоставлении с выполненной механической работой, определяют уровень толерантности организма к физической нагрузке. Минутное потребление кислорода ($\dot{V}O_2$), минутное выведение углекислого газа ($\dot{V}CO_2$), дыхательный коэффициент (отношение $RER = \dot{V}CO_2 / \dot{V}O_2$), минутная вентиляция легких (\dot{V}_E), количество дополнительно выделяемого не метаболического излишка углекислого газа ($Exc_{CO_2} = \dot{V}CO_2 - RER_{покоя} \times \dot{V}O_2$), минутная частота сердечных сокращений — ЧСС, ударов в минуту (уд/мин), концентрация лактата в капиллярной крови являются показателями, описывающими основные физиологические процессы, протекающие в организме при нагрузке. По этим показателям, в дальнейшем, происходит оценка АЭП и ПАНО [5, 6].

Разворачивание физиологических процессов при возрастающей мощности нагрузки состоит из трех фаз, которые в динамике представляют собой кривые физиологических параметров с характерными переходными участками, которые могут быть аппроксимированы кусочно-линейными отрезками. Энергетическое обеспечение в первой фазе осуществляется полностью за счет аэробных метаболических процессов. Первая точка излома, называемая "аэробным порогом", начинает вторую фазу, при которой образование лактата в скелетной мышце превышает его распад и начинает постепенно накапливаться. Третья фаза начинается вторым изломом, который называется "порог анаэробного обмена" и представляет собой критический режим работы, при котором происходит переход от преимущественно аэробного энергообеспечения

мышечной работы к смешанному энергообразованию с нарастающим участием анаэробного гликолиза, сопровождающимся быстрым накоплением лактата в работающих мышцах и крови.

Основной проблемой в определении АЭП и ПАНО является идентификация значений точек перегибов кусочно-линейных кривых физиологических параметров, которые вследствие наличия шума в данных не могут быть определены с высокой точностью (рис. 1, см. третью сторону обложки).

Как правило, обработка данных проводится квалифицированными экспертами в программе MS Excel графическим методом, с построением в ручном режиме регрессионных прямых, и нахождением с их помощью, точек (участков) возрастающего отклонения от линейности графиков физиологических параметров [6].

Для определения АЭП и ПАНО помимо ручного режима применяется специализированное программное обеспечение (ПО), обычно поставляемое в комплекте к газоанализаторам. В таком ПО используется алгоритм расчета ПАНО на основе зависимостей $\dot{V}CO_2$ от $\dot{V}O_2$, пересечения вентиляционных эквивалентов $\dot{V}CO_2 / \dot{V}_E$ и $\dot{V}O_2 / \dot{V}_E$ или зависимости Exc_{CO_2} от величины физической нагрузки, где с помощью метода *V-slope* или его аналогов определяется изменение наклона кривой процесса [7]. Для обработки зашумленных данных предлагается выбрать один из нескольких алгоритмов сглаживания (метод скользящего среднего, построение сплайнов, быстрое преобразование Фурье, фильтр Савицкого—Голея и т. д.), при этом не приводятся практические рекомендации для пользователя по выбору нужного алгоритма и соответствующих ему параметров. Кроме того, при сглаживании и фильтрации происходит потеря определенной части информации, что может приводить к недостоверному определению порогов. Значения ПАНО, полученные разными методами, зависят от выбора алгоритма сглаживания и могут сильно отличаться друг от друга.

Описанные практические сложности ставят задачу создания математически корректных алгоритмов выделения наиболее правдоподобных значений АЭП и ПАНО из зашумленных данных по нескольким физиологическим параметрам и предоставление их эксперту для принятия окончательного решения.

Цели и задачи

Целью работы являлась разработка алгоритма определения аэробного и анаэробного порогов по экспериментальным данным газоанализа при проведении нагрузочного тестирования на эргометре. Работа включала в себя решение следующих задач:

- разработка математического алгоритма автоматизированной оценки АЭП и ПАНО, основанного на робастных регрессионных методах и процедурах оптимизации с одновременным использованием нескольких физиологических кри-

териев достижения АЭП и ПАНО при анализе физиологических параметров, зависящих от времени;

- реализация алгоритма в виде программного модуля;
- сравнение оценок, полученных с помощью разработанного алгоритма, с оценками экспертов-физиологов.

Математический алгоритм

В тестах с возрастающей нагрузкой до отказа временные ряды показателей $V_{CO_2}(t)$, $RER(t)$, $V_E(t)$, $ExcCO_2(t)$ могут быть представлены в следующем уравнением регрессии:

$$y(t) = x(t) + \varepsilon(t),$$

где $x(t)$ — кусочно-линейная функция; $\varepsilon(t)$ — погрешность.

В уравнении регрессии показателя $ExcCO_2$, состоящей из двух частей, присутствует одна точка перегиба. Уравнения регрессии показателей V_{CO_2} , RER , V_E состоят из трех кусочно-линейных участков и имеют две точки перегиба.

Для каждого сопряжения кусочно-линейных участков справедливо выражение

$$y = \begin{cases} y_1 + b_1(x_i - x_1), & x_i < x_2; \\ y_2 + b_2(x_i - x_2), & x_2 \leq x_i < x_3; \\ y_3 + b_3(x_i - x_3), & x_3 \leq x_i \end{cases}$$

где

$$y_2 = y_1 + b_1(x_2 - x_1); \quad y_3 = y_2 + b_2(x_3 - x_2),$$

x_1 — начало выполнения теста; x_2 — АЭП; x_3 — ПАНО.

При проведении нагрузочных тестов присутствует погрешность измерения не только y_i , но и x_i . Для повышения точности определения изгиба кривых используется метод ортогональной регрессии. В этом случае функция суммы квадратов ошибок принимает вид

$$S = \sum_{i=0}^m \frac{(y_1 + b_1(x_i - x_1) - y_i)^2}{b_1^2 + 1} + \sum_{i=m+1}^k \frac{(y_2 + b_2(x_i - x_2) - y_i)^2}{b_2^2 + 1} + \sum_{i=k+1}^N \frac{(y_3 + b_3(x_i - x_3) - y_i)^2}{b_3^2 + 1}.$$

Параметры $b_1, b_2, b_3, y_1, x_2, x_3$ определяются из численного решения задачи оптимизации квази-ньютонским методом [8].

Устойчивость получаемого решения к малым изменениям показателя может быть повышена при использовании специальной процедуры робастного оценивания [9].

Оценка стандартной ошибки наблюдения (x_i, y_i) , получаемая из решения оптимизационной задачи [10], определяется формулой

$$r_i = \frac{y_0 + b(x_i - x_0) - y_i}{\sqrt{b^2 + 1}}.$$

Крайние значения наблюдений (x_i, y_i) корректируются заменой на значения их псевдонаблюдений: если $|r_i| > c\sqrt{S}$, то выполняется корректировка:

$$r'_i = \begin{cases} r_i - c\sqrt{S}, & r_i > 0; \\ r_i + c\sqrt{S}, & r_i < 0; \end{cases}$$

$$x'_i = \frac{r'_i b}{\sqrt{b^2 + 1}} + \frac{(y_i - y_0 + b x_0) b + x_i}{b^2 + 1};$$

$$y'_i = -\frac{1}{b}(x'_i - x_i) + y_i.$$

Константа c регулирует степень робастности. Далее по псевдонаблюдениям (x'_i, y'_i) вычисляются новые значения (\hat{x}_i, \hat{y}_i) подгонки. Действия повторяются до достижения сходимости (x_i, y_i) (рис. 2).

Для значения каждого параметра модели может быть выполнена оценка доверительного интервала (ДИ). Расчет ДИ выполняется с помощью метода бутстреппинга [11]. На основе набора точек одного показателя газоанализа одного нагрузочного теста

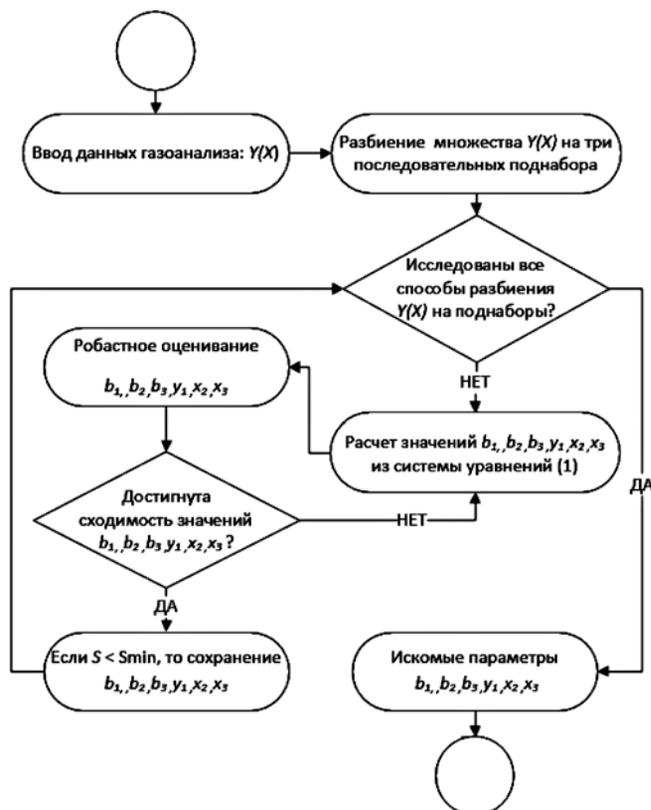


Рис. 2. Алгоритм расчета параметров робастной кусочно-линейной регрессии

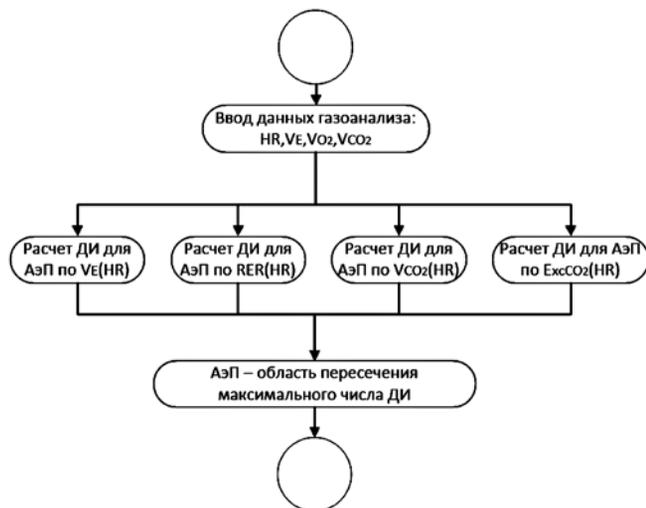


Рис. 3. Блок-схема алгоритма расчета АП на основе применения робастной кусочно-линейной регрессии к нескольким физиологическим параметрам

генерируется множество псевдовыборок того же размера, состоящих из случайных комбинаций исходного набора точек, при этом используется алгоритм случайного выбора с возвращением. Пусть p_i , где $i = 1, \dots, n$, — расчетное значение параметра модели для каждой псевдовыборки. Тогда

$$E(p) = \frac{1}{n} \sum_{i=1}^n p_i;$$

$$SE_{boot}(p) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (p_i - E(p))^2};$$

$$CI_\gamma = [E(p) - t_{\gamma, n} SE_{boot}(p), E(p) + t_{\gamma, n} SE_{boot}(p)],$$

где $E(p)$ — среднее значение расчетного значения параметра p ; $SE_{boot}(p)$ — стандартная ошибка метода бутстреппинга; CI_γ — доверительный интервал оценки параметра p ; t_γ — критическое значение γ -го квантиля распределения Стьюдента $t(\gamma, n - 1)$.

Искомые параметры модели (АэП и ПАНО) также могут быть найдены применением разработанного алгоритма к зависимостям $V_{CO_2}(HR)$, $RER(HR)$, $V_E(RER)$, $Exh_{CO_2}(HR)$, которые получаются исключением из временных рядов показате-

телей газоанализа параметра t . Далее находится область с максимальным перекрытием ДИ по всем парным наборам показателей (рис. 3).

Программный модуль

Описанный алгоритм был реализован в виде программного модуля, на вход которого поступает файл с данными тестирования, а на выходе рассчитываются значения АэП и ПАНО, определенные по нескольким физиологическим критериям, а также их интегральная оценка. Помимо алгоритма, в модуле реализован удобный функционал для определения порогов в ручном режиме экспертами-физиологами. Алгоритм может использоваться для получения первоначальных значений порогов, которые затем, при необходимости, могут быть скорректированы на основе данных по лактату (рис. 4, см. третью сторону обложки).

Применение алгоритма

Рассмотрим пример практического применения алгоритма, взяв реальные данные, полученные во время аэробного тестирования спортсмена на беговой дорожке до отказа.

Спортсмен: возраст 19 лет, рост 179 см, масса 75,3 кг, имеет 1-й спорт. разряд. Начальная скорость дорожки составляла 7 км/ч, с последующим увеличением на 0,1 км/ч каждые 10 с.

На протяжении всей работы, с помощью монитора сердечного ритма "Polar", непрерывно регистрировали ЧСС. Газовый состав выдыхаемого воздуха и показатели внешнего дыхания непрерывно определяли прибором "Metamax 3В" (Германия), который калибровали непосредственно перед проведением каждого исследования. Кровь для определения концентрации лактата брали из дистальной фаланги пальца в последние 20 с каждой 3-й минуты работы.

Процесс реализации метода показан на рис. 5—8, а результаты его применения собраны в табл. 1.

Значения АэП и ПАНО, определенные программным методом, близки к оценкам полученным экспертами-физиологами, которые определяют эти значения пользуясь критериями, описанными в [6]

Таблица 1

Определение АэП и ПАНО по различным критериям

Метод	Параметр	Значение, уд/мин	ДИ 95 %, уд/мин	Экспертная оценка, уд/мин	Ошибка, %	Попадание в ДИ
V_{CO_2}	АэП	145	[144—146]	144	0,7	Да
	ПАНО	179,5	[178—181]	178	0,8	Да
$V_E(HR)$	АэП	144,5	[143—146]	144	0,3	Да
	ПАНО	176	[174—178]	178	1,1	Да
$RER(HR)$	АэП	149	[148—150]	144	3,4	Нет
	ПАНО	181	[180—182]	178	1,6	Нет
$Exh_{CO_2}(HR)$	ПАНО	164,5	[160—169]	178	7,5	Нет

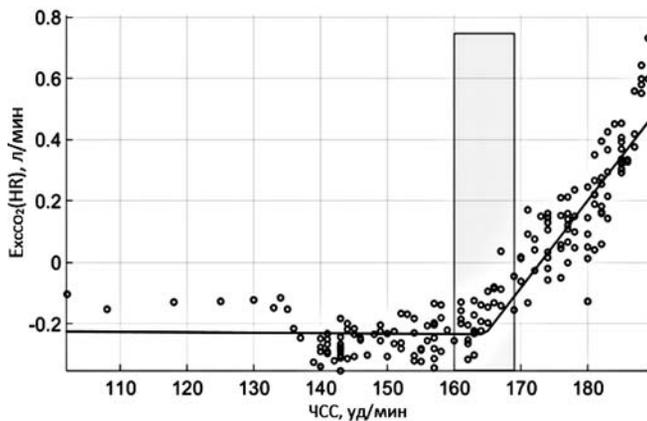


Рис. 5. E_{CO_2} (дополнительное выделение CO_2), ПАНО: [160–169] уд/мин

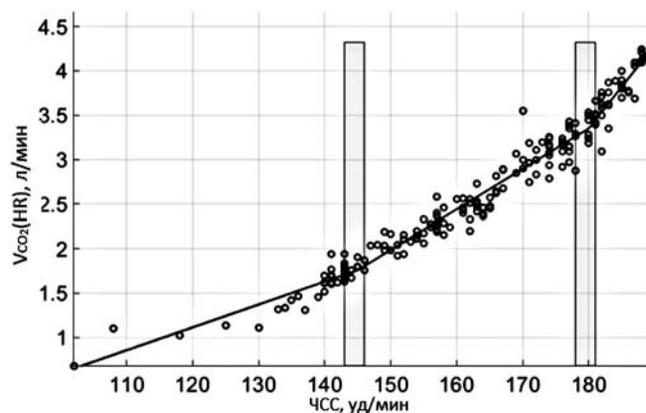


Рис. 7. V_{CO_2} (выделение CO_2), АЭП: [144–146] уд/мин, ПАНО: [178–181] уд/мин

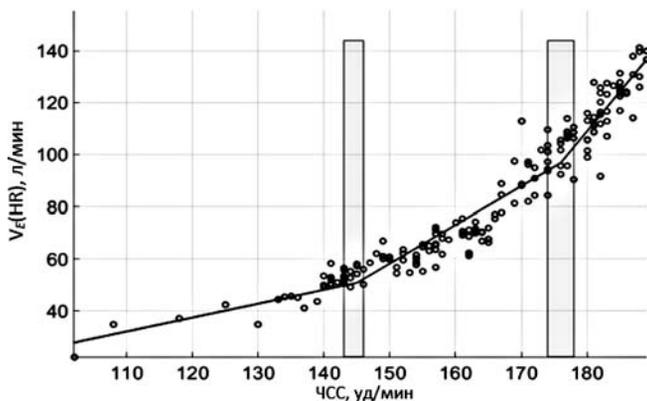


Рис. 6. V_E (вентиляция), АЭП: [143–146] уд/мин, ПАНО: [174–178] уд/мин

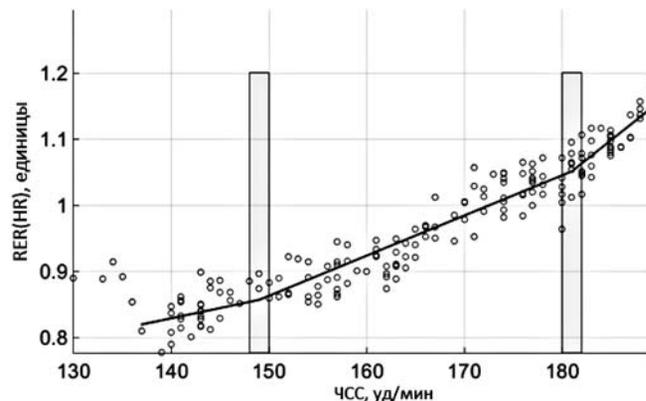


Рис. 8. RER (дыхательный эквивалент), АЭП: [148–150] уд/мин, ПАНО: [180–182] уд/мин

(табл. 1). Для интегральной оценки АЭП и ПАНО по совокупности критериев можно использовать объединение, пересечение и среднее значение доверительных интервалов (табл. 2). Наиболее точным является критерий пересечения интервалов, однако из-за индивидуальных физиологических особенностей полученный интервал может быть пустым. Оценка на основе объединения интервалов может оказаться достаточно широкой, поэтому наиболее универсальным способом являются средние значения по верхним и нижним границам доверительных интервалов используемых критериев.

Таблица 2

Определение АЭП и ПАНО по совокупности критериев

Метод	Параметр	Значение параметров, уд/мин
Объединение ДИ	АЭП	[143–150]
	ПАНО	[160–182]
Пересечение ДИ	АЭП	Пустое множество
	ПАНО	Пустое множество
Среднее значение ДИ	АЭП	[145–147,3]
	ПАНО	[173–177,5]

Заключение

Построен алгоритм обработки, анализа и интерпретации результатов физиологических тестов с возрастающей нагрузкой. Данный алгоритм отличается от имеющихся применением робастных регрессионных методов, отсутствием необходимости предварительного субъективного сглаживания данных пользователем, расчетом доверительных интервалов методом бутстреппинга, который позволяет определять значения аэробного и анаэробного порога сразу по нескольким физиологическим критериям.

Алгоритм реализован в виде программного модуля, на вход которого поступают данные газоанализа, а после расчета основных показателей проводится визуализация основных кривых с выделением физиологически важных зон.

Разработанный программный модуль, апробированный на данных нагрузочного тестирования спортсменов-легкоатлетов, специализирующихся в беге на средние дистанции, получил высокую оценку экспертов-физиологов, снизив время, затрачиваемое на определение пороговых значений и повысив при этом точность.

Список литературы

1. Павлов В., Линде Е., Орджоникидзе З., Николаев В., Деев В. Сравнительный анализ нагрузочного тестирования на различных видах эргометров // Наука в олимпийском спорте. 2014. № 2. С. 46—50.
2. Тавровская Т. В. Велоэргометрия. Практическое пособие для врачей. Санкт-Петербург: Научно-производственное предприятие "Нео", 2007. 134 с.
3. Белоцерковский З. Б. Эргометрические и кардиологические критерии физической работоспособности у спортсменов. М.: Советский спорт 2005. 312 с.
4. Burnley M., Doust J. H., Ball D., Jones A. M. Effects of prior heavy exercise on V_{CO_2} kinetics during heavy exercise are related to changes in muscle activity // Journal of Applied Physiology. 2002. T. 93. № 1. С. 167—174.
5. Сонькин В. Д., Тамбовцева Р. В. Развитие мышечной энергетики и работоспособности в онтогенезе. М.: Книжный дом "ЛИБРОКОМ", 2011. 368 с.
6. Бреслав И. С., Волков Н. И., Тамбовцева Р. В. Дыхание мышечная активность в спорте: руководство для изучающих физиологию человека. М.: Советский спорт, 2013. С. 67—172.
7. Epistemic Mindworks WinBreak Automated software program for the detection of the gas exchange threshold: User Guide, 2003. P. 35—64.
8. Поляк Б. Т. Введение в оптимизацию. М.: Наука. Главная редакция физико-математической литературы, 1983. С. 77—80.
9. Хьюбер П. Робастность в статистике. М.: Мир, 1984. С. 198—229.
10. Boggs P. T., Spiegelman C. H., Donaldson J. R. A Computational Examination of Orthogonal Distance Regression // Journal of Econometrics. 1988. Vol. 38, N. 1—2. P. 169—201.
11. Эфрон Б. Нетрадиционные методы многомерного статистического анализа. М.: Финансы и статистика, 1988. С. 175—213.

A. V. Golov^{1, 2}, Graduate Student, e-mail: golov.andrey@hotmail.com,

E. A. Timme^{2, 3}, Researcher, e-mail: alpdem@yandex.ru,

A. V. Kozlov^{2, 4}, Chief specialist, Graduate student, e-mail: 89165363985a@mail.ru

¹Moscow Institute of Physics and Technology (State University), Moscow, Russia

²Moscow Center of Advanced Sports Technologies, Moscow, Russia

³State Research Center of Russian Federation — Institute of Biomedical Problems of the Russian Academy of Sciences, Moscow, Russia

⁴Russian State University of Physical Education, Sport, Youth and Tourism, Moscow, Russia

The Algorithm of Automated Estimation of Human Performance Parameters under Load Tests

Load physiological tests are universal method of estimation individual performance parameters that characterize source of energy of muscular work. The main purpose of load testing is estimation the aerobic and the anaerobic thresholds. The main problem of estimation of the aerobic and the anaerobic thresholds is processing of experimental results, which is held for the most part in manual mode and interpretation of data is often too difficult. It is not always possible to clearly identify important physiological information duo to noise, outliers, a significant dependence on the initial test conditions and individual modes of roll out of physiological processes. Estimations of the aerobic and the anaerobic thresholds obtained by different criteria can vary greatly and therefore the attempts of their association with each other give rise to more ambiguity.

The aim of this work is to develop mathematical algorithm of determination of the aerobic and the anaerobic thresholds based on gas analysis experimental data during tests with increasing load power protocols on different types of ergometers. Developed algorithm based on robust regressions methods, procedures of optimization, confidence interval computation by bootstrapping method. The algorithm uses data of multiple physiological parameters such as oxygen consumption, release carbon dioxide, respiratory coefficient, pulmonary ventilation, heart rate and is implemented as a software module. Comparison of estimates obtained by the developed method with estimates of experts-physiologists.

The software module has been developed and tested on data of endurance sports athletes. Obtained values of aerobic threshold and anaerobic threshold are consistent with expert's estimations. Therefore, the method can be successfully used in everyday practice of laboratory testing of athletes.

Keywords: functional load test, oxygen consumption, aerobic threshold, anaerobic threshold, ergometers, load test protocol, performance level, robust piecewise regression, bootstrapping method

References

1. Pavlov V., Linde E., Ordzhonikidze Z., Nikolaev V., Deev V. Sravnitel'nyj analiz nagruzochnogo testirovaniya na razlichnyh vidah ehrgometrov, *Nauka v olimpijskom sporte*, 2014, no. 2, pp. 46—50 (in Russian).
2. Tavrovskaya T. V. *Veloehrgometriya. Prakticheskoe posobie dlya vrachej*. Sankt-Peterburg: Nauchno-proizvodstvennoye predpriyatie "Neo", 2007. 134 p. (in Russian).
3. Belocerkovskij Z. B. *Ergometricheskie i kardiologicheskie kriterii fizicheskoj rabotosposobnosti u sportsmenov*. Moscow: Sovetskij sport 2005. 312 p. (in Russian).
4. Burnley M., Doust J. H., Ball D., Jones A. M. Effects of prior heavy exercise on V_{CO_2} kinetics during heavy exercise are related to changes in muscle activity, *Journal of Applied Physiology*, 2002, vol. 93, no. 1, pp. 167—174.
5. Sonkin V. D., Tambovtseva R. V. *Development of muscular energy and health in ontogenesis*. Moscow: Book house "LIBROKOM", 2011. 368 p. (in Russian).
6. Breslav I. S., Volkov N.I., Tambovtseva R. V. *Dyhanie myshechnaya aktivnost' sporte: rukovodstvo dlya izuchayushchih fiziologiyu cheloveka*, Moscow, Sovetskij sport, 2013, pp. 67—172 (in Russian).
7. Epistemic Mindworks WinBreak Automated software program for the detection of the gas exchange threshold, User Guide, 2003, pp. 35—64.
8. Polyak B. T. *Vvedenie v optimizaciyu*, Moscow, Nauka. Glavnaya redakciya fiziko-matematicheskoy literatury, 1983, pp. 77—80 (in Russian).
9. H'yuber P. *Robastnost' v statistike*, Moscow, Mir, 1984, pp. 198—229 (in Russian).
10. Boggs P. T., Spiegelman C. H., Donaldson J. R. A Computational Examination of Orthogonal Distance Regression, *Journal of Econometrics*, 1988, vol. 38, no 1—2, pp. 169—201.
11. Efron B. *Netradicionnye metody mnogomernogo statisticheskogo analiza*, Moscow, Finansy i statistika, 1988, pp. 175—213 (in Russian).

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.932.2

С. В. Белим, д-р физ.-мат. наук, проф.,

зав. кафедрой информационной безопасности, e-mail: belimsv@omsu.ru,

С. Б. Ларионов, аспирант кафедры информационной безопасности, e-mail: me@stas-larionov.ru

Омский государственный университет им. Ф. М. Достоевского, Омск

Алгоритм выявления поврежденных пикселей изображения на основе выявления сообществ на графах

Цель данной работы — анализ изображений, поврежденных импульсным шумом. Решается задача выявления поврежденных пикселей. Данная задача возникает на первом этапе улучшения зашумленных изображений. В дальнейшем, зная положение поврежденных пикселей, легко аппроксимировать их исходные значения. В работе предложен алгоритм выявления пикселей, поврежденных импульсным шумом, основанный на сегментации изображения с помощью поиска сообществ на графах. Изображение, содержащее импульсный шум, представляется в виде неориентированного взвешенного графа. Для определения весов ребер используется экспоненциальная функция, зависящая от цветовых компонент соседних пикселей. Полученный граф разбивается на сообщества. Поврежденные пиксели находятся как сообщества, состоящие из одной вершины. На основе этого подхода строится алгоритм, имеющий линейную трудоемкость. В предлагаемом алгоритме для каждого пикселя изображения исследуется целесообразность его объединения в один сегмент со всеми ближайшими соседями. Пиксель считается поврежденным, если объединение его в сообщество с ближайшими соседями не является выгодным. Проведен компьютерный эксперимент как на искусственных изображениях, так и на фотографиях. Для всех изображений моделируется импульсный шум с помощью генератора псевдослучайных чисел. Были проанализированы изображения с уровнем зашумления от 10 до 70 %. Компьютерный эксперимент показал, что предложенный метод обладает преимуществами по сравнению с другими методами при высоком уровне зашумления изображения. При низких уровнях зашумления (до 30 %) результаты предложенного алгоритма сопоставимы с аналогами. При больших уровнях зашумления этот алгоритм имеет значительно большую эффективность по сравнению с другими аналогичными алгоритмами. Следует отметить более низкий уровень ложных срабатываний предложенного алгоритма по сравнению с аналогами. При этом с ростом зашумления уровень ложных срабатываний убывает, что выгодно отличает предложенный в работе алгоритм.

Ключевые слова: анализ изображений, импульсный шум, улучшение изображения, выявление сообществ

Введение

Проблема выявления поврежденных пикселей на изображениях относится к задачам обучения без учителя. Большинство алгоритмов, решающие данную проблему, основаны на исследовании некоторой малой окрестности каждого пикселя, называемого окном [1, 2]. Из сравнения пикселя, расположенного в центре окна, с ближайшими соседями делается вывод о его принадлежности к шуму. Основное отличие данного типа алгоритмов друг от друга состоит в выборе размера окна и метода принятия решения о поврежденности пикселя [3–5]. Поиск всех поврежденных пикселей осуществляется одним проходом по всем точкам изображения, в связи с чем алгоритмы имеют линейную трудоемкость. Эффективность данных алгоритмов сильно зависит от правильного выбора параметров, уста-

навливаемых пользователем в ручном режиме. Так, алгоритм SD-ROM [1] требует настройки четырех параметров, для которых известны только рекомендуемые интервалы. Различные наборы параметров на одном и том же изображении приводят к эффективности работы алгоритма от 62 до 96 %. Также следует отметить высокий процент ложных срабатываний указанных алгоритмов.

Альтернативный подход связан с принятием решения о поврежденности пикселя на основе анализа изображения в целом. К таким подходам можно отнести алгоритмы, основанные на выявлении ассоциативных правил для пикселей изображения [6, 7], методах поддержки принятия решений [8], а также основанные на методах сегментации изображений [9, 10]. Данные алгоритмы характеризуются меньшим числом настраиваемых параметров, высокой эффективностью и низким процентом ложных

срабатываний. К недостаткам алгоритмов можно отнести их более высокую трудоемкость по сравнению с методами, основанными на подвижном окне.

Один из методов сегментации изображений, получивший развитие в последнее время, основан на представлении изображения в виде взвешенного графа с последующим выявлением сообществ (*community*) на данном графе [11–13]. Причем выявление сообществ может быть реализовано как алгоритм выращивания областей на исходном изображении, характеризующийся квадратичной трудоемкостью. Можно ожидать, что, как и в других методах сегментации, пиксели, испорченные импульсным шумом, будут выделяться в отдельные кластеры, состоящие из одной вершины.

Цель данной статьи — разработка и тестирование алгоритма поиска на изображении пикселей, поврежденных импульсным шумом.

Постановка задачи и алгоритм решения

Импульсный шум на цифровых изображениях приводит к изменению цвета отдельных случайно расположенных пикселей. Причем новый цвет пикселя также имеет случайное значение. Отношение числа испорченных пикселей к общему числу пикселей изображения принято называть процентом зашумления, в дальнейшем будем обозначать его через p .

Общая модель зашумленного изображения может быть описана следующим образом. Пусть исходное изображение задано в виде матрицы цветов \mathbf{A}_{ij} . Диапазон значений элементов матрицы $[0, m - 1]$ зависит от глубины палитры. Пусть поврежденное изображение задается матрицей цветов \mathbf{B}_{ij} . Введем матрицу \mathbf{R}_{ij} , определяющую положение пикселей, измененных импульсным шумом. Каждый элемент матрицы \mathbf{R}_{ij} с вероятностью p отличен от нуля и с вероятностью $1 - p$ нулевой. Значение ненулевых элементов матрицы повреждений также задается случайным образом в интервале $[0, m - 1]$. Значение матрицы поврежденного изображения задается соотношением

$$\mathbf{B}_{ij} = \begin{cases} \mathbf{R}_{ij}, & \mathbf{R}_{ij} > 0, \\ \mathbf{A}_{ij}, & \mathbf{A}_{ij} = 0. \end{cases}$$

Поиск поврежденных пикселей сводится к определению матрицы \mathbf{R}_{ij} исходя из анализа матрицы \mathbf{B}_{ij} при неизвестной матрице \mathbf{A}_{ij} . Может быть поставлена более широкая задача поиска матрицы \mathbf{A}_{ij} из анализа матрицы \mathbf{B}_{ij} , которая соответствует проблеме устранения импульсного шума на цифровых изображениях.

Будем рассматривать изображения размерами $N \times M$ пикселей. Положение отдельного пикселя изображения определяется двумя пространственными координатами (x, y) , где x — целое число из отрезка $[0, N - 1]$, y — целое число из отрезка $[0, M - 1]$. Бу-

дем рассматривать цветные изображения. В этом случае пикселю с координатами (x, y) будут соответствовать три цветовые составляющие: $r(x, y)$ — интенсивность красного цвета; $g(x, y)$ — интенсивность зеленого цвета; $b(x, y)$ — интенсивность синего цвета. При обработке монохромного изображения будем считать, что интенсивность двух из трех составляющих цвета нулевая.

Построим неориентированный взвешенный граф G , соответствующий изображению, выбрав в качестве вершин пиксели изображения. Соединим ребрами ближайшие соседние пиксели. Ближайшими соседними пикселями v_i и v_j будем считать пиксели, каждая из координат которых отличается не больше, чем на единицу: $(|x_i - x_j| \leq 1) \vee (|y_i - y_j| \leq 1)$. Каждому ребру графа припишем вес, характеризующий меру отличия цветов пикселей. Как показано в работе [14], необходимыми свойствами обладает экспоненциальная функция. Будем вычислять вес ребра между вершинами $v_i = (x_i, y_i)$ и $v_j = (x_j, y_j)$ с помощью формулы

$$d(v_i, v_j) = \exp\left(-\frac{1}{h} \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}\right).$$

Здесь $r_i = r(v_i, v_j)$, $g_i = g(v_i, v_j)$, $b_i = b(v_i, v_j)$. Параметр h определяет значение изменения цвета, которое детектируется как переход к соседнему сегменту. Данный параметр определяется пользователем и является общим для всего изображения.

При сегментации изображения основной целью является поиск множеств пикселей, схожих по цвету. Мерой схожести служит "цветовое расстояние" между пикселями $d(v_i, v_j)$. Сегментация изображения равносильна разбиению графа G на подграфы, вершины которых между собой более близки, чем с вершинами других подграфов. Такие подграфы принято называть сообществами (*community*). Для оценки качества разбиения графа на сообщества наибольшее распространение получила функция модульности Ньюмана [15, 16]. Если провести полное разбиение графа G , соответствующего изображению, на сообщества, то некоторые из сообществ будут состоять из одной вершины. Пиксели, соответствующие вершинам, составляющим сообщество из одного элемента, значительно отличаются по цвету от своего окружения и, как следствие, с высокой вероятностью повреждены шумом. Поставим задачу поиска на графе G , соответствующем изображению, одновершинных сообществ.

Зададим для графа G матрицу весов \mathbf{E} следующим образом. Диагональные элементы E_{ii} показывают вес вершин. На нулевом шаге все вершины имеют нулевой вес. Недиагональные элементы E_{ij} ($i \neq j$) равны весу ребра между вершинами v_i и v_j . Так как мы рассматриваем только ближайшие соседние пиксели, то из каждой вершины выходит не более восьми ребер и на нулевом шаге в каждой

строке будет не более восьми ненулевых значений. Граф неориентированный, поэтому матрица \mathbf{E} будет симметричной относительно главной диагонали. Следовательно, максимальный объем требуемой памяти составляет $4MN$ значений весов ребер. Для вычисления функции модульности перейдем к приведенному виду матрицы весов $\mathbf{e} = \mathbf{E}/\mathbf{m}$, где

$$\mathbf{m} = \sum_{i,j=1}^{MN} \mathbf{E}_{ij}. \text{ В приведенной матрице весов элемент}$$

\mathbf{e}_{ij} показывает долю веса заданного ребра в общем весе графа. В дальнейшем под матрицей весов будем понимать именно приведенный вид. Легко видеть, что

$$\sum_{i,j=1}^{MN} \mathbf{e}_{ij} = 1.$$

Значение модульности, численно определяющее качество разбиения графа, в общем виде [15, 16] определяется выражением

$$Q(G) = \sum_{i=1}^K \mathbf{e}_{ij} - \sum_{i=1}^K a_i b_i,$$

где K — число вершин графа; a_i — приведенная

исходящая степень вершины $v_i \left(a_i = \sum_{j=1, j \neq i}^K \mathbf{e}_{ij} \right)$;

b_i — приведенная входящая степень вершины v_i

$\left(b_i = \sum_{j=1, j \neq i}^K \mathbf{e}_{ji} \right)$. Рассматриваемый граф является

неориентированным, поэтому входящая и исходящая степени для всех вершин одинаковы ($a_i = b_i$; $i = 1, \dots, K$). Функция модульности принимает более простой вид:

$$Q(G) = \sum_{i=1}^K \mathbf{e}_{ij} - \sum_{i=1}^K a_i^2.$$

Для поиска сообществ на графе будем использовать процедуру образования стяжек. Под стяжкой понимается преобразование, при котором некоторый подграф H графа G заменяется одной вершиной v_H . Если какая-то из вершин подграфа H была связана ребром с вершиной v из подграфа $G \setminus H$, то вершина v_H также будет связана ребром того же веса с вершиной v . Вес новой вершины будет равен сумме весов ребер и вершин, входящих в подграф H . Новый граф обозначим G_H . Подграф H будем считать сообществом, если $Q(G_H) > Q(G)$. Следует отметить, что при образовании стяжки уменьшается число вершин графа K . Нашей задачей стоит поиск вершин графа, не входящих ни в одно более крупное сообщество. Выявление таких вершин будем осуществлять с помощью следующего алгоритма.

Шаг 1. Осуществляем последовательный проход по всем пикселям изображения.

Шаг 2. Для каждого пикселя v поочередно рассматриваем ближайшие соседние пиксели $v^{(i)}$ ($i = 1, \dots, 8$). Рассматриваем подграфы, состоящие из двух вершин v и $v^{(i)}$ ($i = 1, \dots, 8$), и пытаемся объединить их в сообщество. При каждом объединении вычисляем изменение функции модульности ΔQ_i ($i = 1, \dots, 8$).

Шаг 3. Если при объединении данной вершины в сообщество со всеми ближайшими соседями изменение функции модульности отрицательно ($\Delta Q_i < 0$, $i = 1, \dots, 8$), то соответствующий пиксель считаем поврежденным.

На выходе алгоритма получим список пикселей, существенно отличающихся по цвету от своего окружения. С большой вероятностью данные пиксели можно считать поврежденными, однако они могут оказаться и особенностями исходного изображения. Если на третьем шаге алгоритма запоминать координаты поврежденных пикселей, то на выходе получим матрицу \mathbf{R}_{ij} .

Изменение функции модульности можно достаточно быстро вычислить из текущих характеристик графа, соответствующего изображению. При объединении вершин v_i и v_j изменение модульности будет иметь вид

$$\Delta Q = 2(\mathbf{e}_{ij} - a_i a_j).$$

Очевидно, что предложенный алгоритм имеет линейную трудоемкость по числу пикселей.

Компьютерный эксперимент

Компьютерный эксперимент проводили на искусственных изображениях и фотографиях различных объектов. На вход алгоритма подавали изображения размером $N = 512$, $M = 512$ пикселей. После этого с помощью линейного конгруэнтного генератора генерировали координаты pNM точек, моделирующих случайный импульсный шум. Уровень зашумления p изменялся в интервале от 0 до 70 %. Рассматривали 50 различных изображений одного типа. Для каждого изображения и каждого уровня зашумления проводили по 100 независимых экспериментов. В каждом эксперименте генерировались испорченные пиксели и сохранялись их координаты. После этого осуществляли поиск испорченных пикселей и результаты сравнивали с исходными данными. Для численной оценки результатов вычисляли две величины: эффективность работы алгоритма

$$eff = \frac{N_0}{pNM},$$

где N_0 — число верно идентифицированных испорченных пикселей;

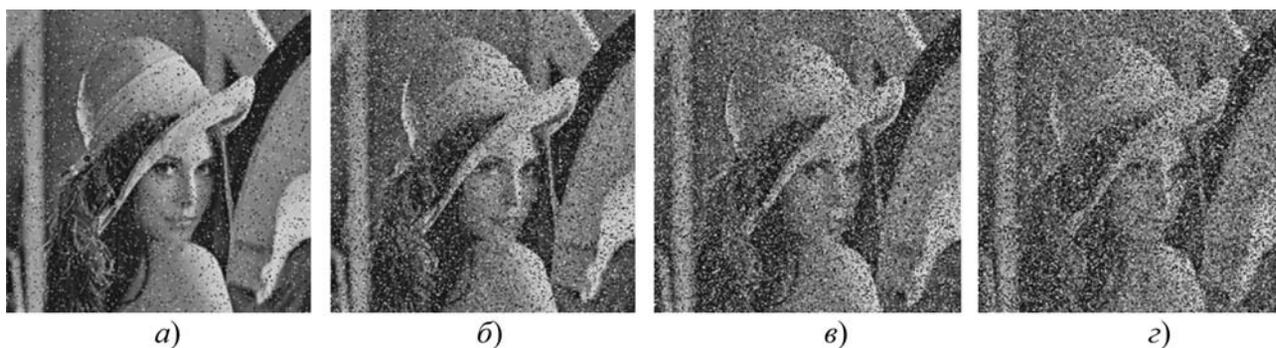


Рис. 1. Тестовые фотографические изображения с различным уровнем зашумления: а — $p = 10\%$; б — $p = 30\%$; в — $p = 50\%$; г — $p = 70\%$

уровень ложных срабатываний

$$err = \frac{N_e}{N},$$

где N_e — число ошибочно идентифицированных испорченных пикселей; N — общее число пикселей, определенное алгоритмом как испорченные.

В первом эксперименте в качестве изображения была выбрана прямоугольная область с равномерной заливкой. В этом случае эффективность работы алгоритма при $p = 10\%$ в среднем была равна $eff = 99,98\%$ при уровне ложных срабатываний $err = 0,11\%$. Отличие эффективности от стопроцентного значения обусловлено тем, что цвет испорченных пикселей выбирался случайно и для некоторых из них оказывался близким к цвету заливки. Уровень эффективности обнаружения испорченных пикселей остается высоким и при больших уровнях зашумления. Так при $p = 70\%$ эффективность составляет $eff = 99,97\%$ при уровне ложных срабатываний $err = 6,20\%$. Рост уровня ложных срабатываний объясняется тем, что при увеличении уровня зашумления вероятность того, что испорченные пиксели окажутся ближайшими соседями становится высокой.

Компьютерный эксперимент для искусственных изображений геометрических фигур с областями градиентной заливки показал, что эффективность работы алгоритма для уровней зашумления от $p = 10\%$ до $p = 70\%$ понижается незначительно от $eff = 99,97\%$ до $eff = 99,95\%$. Уровень ложных срабатываний понижается с ростом зашумления от $err = 7,81\%$ до $err = 4,15\%$. Детальный сравнительный анализ результатов показывает, что наибольший вклад в уровень ложных срабатываний вносят резкие границы, разделяющие области изображения.

Далее эксперименты проводили для широко известного изображения "Lena". Тестовые изображения с различным уровнем зашумления представлены на рис. 1. Для каждого уровня зашумления выполняли по 100 независимых экспериментов. Ком-

пьютерный эксперимент на различных изображениях показал, что наилучших результатов удается добиться при $h = 0,5$.

На рис. 2 представлен график зависимости средней эффективности работы алгоритма и среднего уровня ложных срабатываний от уровня зашумления изображения "Lena".

Как хорошо видно из рис. 2, эффективность поиска поврежденных пикселей убывает очень медленно с ростом зашумления изображения и даже для $p = 50\%$ составляет $eff = 83,85\%$. Уровень ложных срабатываний также убывает. Это связано с тем, что уровень ложных срабатываний является относительной величиной. Если посмотреть на число неверно определенных пикселей по абсолютному значению, то оно остается практически неизменным при всех уровнях зашумления. Ошибочно определенные пиксели являются мелкими деталями изображения.

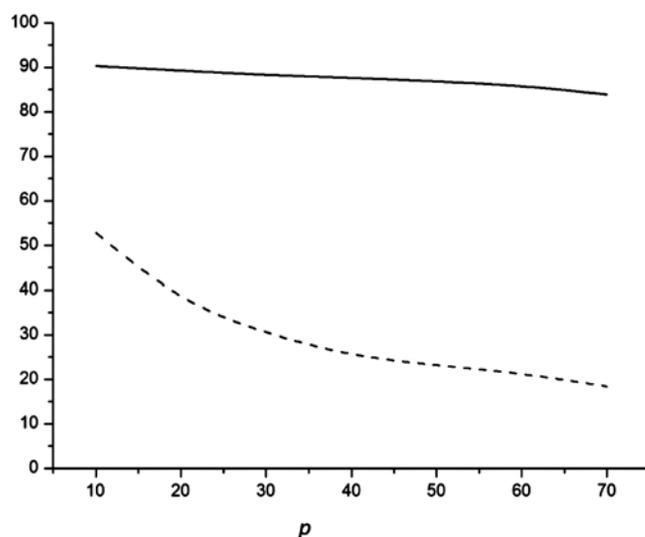


Рис. 2. Зависимость эффективности обнаружения испорченных пикселей и процента ложных срабатываний от уровня зашумления изображения. Сплошной линией показан график для эффективности, штриховой линией — для уровня ложных срабатываний

Анализ результатов и выводы

Предложенный метод выявления пикселей, поврежденных импульсным шумом, имеет достаточно высокую эффективность, даже для сильно зашумленных изображений. Причем число пикселей, ошибочно отнесенных к поврежденным, растет очень медленно с увеличением зашумления, что приводит к уменьшению уровня ложных срабатываний.

Проведем сравнение с результатами других работ. Наиболее распространенный алгоритм SD-ROM позволяет обнаруживать испорченные пиксели в среднем с эффективностью 71 % и уровне ложных срабатываний 5 % при интенсивности шума 10 % [1]. Этот же метод может демонстрировать эффективность до 96 % при уровне зашумления 30 %, но на шумах типа "соль и перец". Следует отметить, что задача анализа изображений с шумом "соль и перец" является более простой, так как в них интенсивность цвета испорченных пикселей имеет либо максимальное, либо минимальное значение. Причем достижение максимальной эффективности возможно лишь при определенном наборе четырех параметров, подбираемых вручную. При увеличении уровня шума уровень ложных срабатываний в алгоритме SD-ROM растет очень быстро и уже при интенсивности шума 30 % достигает 73 %.

Аналогичные алгоритмы, основанные на анализе локального окружения каждого пикселя, демонстрируют близкие результаты. Так, метод, использующий построение ассоциативных правил для наборов соседних пикселей, при интенсивности шума 30 %, показывает эффективность 77 % и уровень ложных срабатываний 35 % [6, 7]. Алгоритм, основанный на методе анализа иерархий, позволяет обнаруживать импульсный шум с эффективностью 85 % и ложными срабатываниями 55 % [8]. Существенным недостатком данных методов является анализ достаточно небольшой области изображения вокруг каждого пикселя, который не позволяет выявлять и анализировать протяженные области на изображении. Частично данный недостаток был устранен в методе выявления импульсного шума, основанном на кластеризации изображений [9, 10], для которого эффективность обнаружения испорченных пикселей 82 %, а уровень ложных срабатываний 23 % при зашумлении 10 %. При увеличении процента испорченных пикселей эффективность падает и при 30 % зашумления составляет 68 %, а уровень ложных срабатываний 11 %.

Таким образом, предложенный в данной работе алгоритм позволяет получать заметный выигрыш как в эффективности обнаружения испорченных пикселей, так и в уровне ложных срабатываний при значительном зашумлении изображения, выше 30 %.

Список литературы

1. **Abreu E., Lightstone M., Mitra S. K., Arakawa S. K.** A new efficient approach for the removal of impulse noise from highly corrupted images // *IEEE Transactions on Image Processing*, IEEE Transactions on. 1996. Vol. 5. P. 1012–1025.
2. **Garnett R., Huegerich T., Chui C., He W.** A Universal Noise Removal Algorithm with an Impulse Detector // *IEEE Trans Image Process.* 2005. N. 14 (11). P. 1747–1754.
3. **Сорокин С. В., Щербаков М. А.** Реализация SD-ROM фильтра на основе концепции нечеткой логики // *Известия высших учебных заведений. Поволжский регион.* 2007. № 3. С. 56–65.
4. **Красовский Г. Я., Усс М. Л.** Фильтрация изображений, искаженных импульсными помехами точечного и строчного типа, на основе систем итерированных функций // *Радиоэлектронні і комп'ютерні системи.* 2003. В. 2. С. 47–55.
5. **Chan R., Ho C., Nikolova M.** Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization // *IEEE Trans. Image Proc.* 2005. Vol. 14, N 10. P. 1479–1485.
6. **Белим С. В., Майоров-Зильбернагель А. О.** Алгоритм поиска поврежденных пикселей и удаления импульсного шума на изображении с использованием метода ассоциативных правил // *Наука и образование: электронное научно-техническое издание.* 2014. № 12. URL: <http://technomag.bmstu.ru/doc/744983.html>
7. **Белим С. В., Майоров-Зильбернагель А. О.** Восстановление изображений со статическими пропусками на основе метода ассоциативных правил // *Вестник компьютерных и информационных технологий.* 2014. № 12. С. 18–23.
8. **Белим С. В., Селиверстов С. А.** Использование метода анализа иерархий для выявления импульсного шума в графических объектах // *Информационные технологии.* 2015. № 4. С. 251–258.
9. **Белим С. В., Кутлуни П. Е.** Выявление поврежденных пикселей на изображении с помощью алгоритма кластеризации // *Вестник компьютерных и информационных технологий.* 2016. № 3. С. 3–10.
10. **Белим С. В., Кутлуни П. Е.** Выделение контуров на изображении с помощью алгоритма кластеризации // *Компьютерная оптика.* 2015. Т. 39, № 1. С. 119–124.
11. **Pourian P., Karthikeyan S., Manjunath B. S.** Weakly supervised graph based semantic segmentation by learning communities of image-parts // *The IEEE International Conference on Computer Vision (ICCV).* 2015. P. 1359–1367.
12. **Nair R. S., Vineetha K. V.** Modularity Based Color Image Segmentation // *IJREEICE.* 2016. Vol. 3, Special Is. 1. P. 109–113.
13. **Mourchid Y., Hassouni M. E., Cherif H. A.** New Image Segmentation Approach using Community Detection Algorithms // *International Journal of Computer Information Systems and Industrial Management Applications.* 2016. Vol. 8. P. 195–204.
14. **Browet A., Absil P.-A., Van Dooren P.** Community Detection for Hierarchical Image Segmentation // *IWCIA.* 2011. LNCS 6636. P. 358–371.
15. **Newman M. E.** Analysis of weighted networks // *Physical Review E.* 2004. Vol. 70, N. 5. P. 056131.
16. **Cluset A., Newman M. E. J., Moore C.** Finding community structure in very large networks // *Physical Review E.* 2004. Vol. 70, N. 6. P. 066111.

S. V. Belim, D. Sc. Professor, Head of Department, e-mail: sbelim@mail.ru,
S. B. Larionov, Postgraduate Student, e-mail: me@stas-larionov.ru
Omsk State University n. a. F. M. Dostoevski, Omsk, Russian Federation

The Algorithm of Broken Pixel Detection Based on Community Detection in Graphs

The goal of this article is to analyze images that damaged using impulse noise. The first issue that need to be solved is searching of damaged pixels. This issue is causing on first step of noisy image restoring. Later one can restore source values of pixels based on already known positions of damaged pixels using approximation. This article suggests an algorithm of pixels detection, broken by impulse noise using image segmentation, based on community detection in graphs. The image that contains noise is representing as non-oriented, weighted graph. There is an exponential function that depends on color components of neighbour pixels is using to determine weights of edges. The resulting graph is splitting on communities. Damaged pixels are communities that contain only one vertex. Based on this approach one can build an algorithm with linear complexity. Suggested algorithm checking an expediency of combination each pixel of image with it nearest neighbors. Pixel assumed broken if it combination with nearest neighbors is not profitable. The computer experiment was carried out both on artificial images and photos. For each images impulse noise was generating using generator of pseudo-random numbers. Noise level of analyzed images was from 10 % to 70 %. Computer experiment showed that suggested method has some advantages compared to others when noise level is high. When noise level is low, the results of suggested algorithm are comparable to analogs. Suggested method has much greater efficiency compared to others when noise level is high. It should be pointed out that algorithm has lower level of false positives then analogs. Additionally, with increasing of noise level, count of false positives decreases which distinguishes the proposed algorithm.

Keywords: image analyzing, impulse noise, image restoring, community detection, community

References

1. Abreu E., Lightstone M., Mitra S. K., Arakawa S. K. A new efficient approach for the removal of impulse noise from highly corrupted images, *IEEE Transactions on Image Processing*, *IEEE Transactions on*, 1996, vol. 5, pp. 1012–1025.
2. Garnett R., Huegerich T., Chui C., He W. A Universal Noise Removal Algorithm with an Impulse Detector, *IEEE Trans Image Process*, 2005, vol. 14, no. 11, pp. 1747–1754.
3. Sorokin S. V., Shherbakov M. A. Realizacija SD-ROM fil'tra na osnove koncepcii nechetkoj logiki. (The SD-ROM filter implementation based on the fuzzy logic concept), *Izvestija vysshih uchebnyh zavedenij. Povolzhskij region*, 2007, no. 3, pp. 56–65 (in Russian).
4. Krasovskij G. Ja., Uss M. L. Fil'tracija izobrazhenij, iskazennyh impul'snymi pomehami tochechnogo i strochnogo tipa, na osnove sistem iterirovanyh funkcij. (Filtration of broken by string and dotted impulse images based on system of iterated functions), *Radioelektronni i komp'yuterni sistemi*, 2003, vol. 2, pp. 47–55 (in Russian).
5. Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, *IEEE Trans. Image Proc.*, 2005, vol. 14, no. 10, pp. 1479–1485.
6. Belim S. V., Majorov-Zil'bernagel' A. O. Algoritm poiska povrezhdennyh pikselej i udaleniya impul'snogo shuma na izobrazhenijah s ispol'zovaniem metoda asociativnyh pravil (The algorithm of broken pixel detection and impulse noise removal based on associative rules), *Nauka i obrazovanie: jelektronnoe nauchno-tehnicheskoe izdanie*, 2014, no. 12, URL: <http://technomag.bmstu.ru/doc/744983.html> (in Russian).
7. Belim S. V., Majorov-Zil'bernagel' A. O. Vosstanovlenie izobrazhenij so staticheskimi propuskami na osnove metoda asociativnyh pravil. (Recovering of images with static gaps based on the method of associative rules), *Vestnik komp'yuternyh i informacionnyh tehnologij*, 2014, no. 12, pp. 18–23 (in Russian).
8. Belim S. V., Seliverstov S. A. Ispol'zovanie metoda analiza ierarxij dlya vyavleniya impul'snogo shuma v graficheskix ob'ektax. (Using of hierarchy analysis method for impulse noise recognizing in graphical objects), *Informacionnye texnologii*, 2015, no. 4, pp. 251–258 (in Russian).
9. Belim S. V., Kutlunin P. E. Vyyavlenie povrezhdennyh pikselej na izobrazhenii s pomoshh'yu algoritma klasterizacii (Broken pixels detection in images based on the clusterization algorithm), *Vestnik komp'yuternyh i informacionnyh tehnologij*, 2016, no. 3, pp. 3–10 (in Russian).
10. Belim S. V., Kutlunin P. E. Vydelenie konturov na izobrazheniyax s pomoshh'yu algoritma klasterizacii. (Contours detection in images based on the clusterization algorithm), *Komp'yuternaya optika*, 2015, vol. 39, no. 1, pp. 119–124 (in Russian).
11. Pourian P., Karthikeyan S., Manjunath B. S. Weakly supervised graph based semantic segmentation by learning communities of image-parts, *The IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1359–1367.
12. Nair R. S., Vineetha K. V. Modularity Based Color Image Segmentation, *IJIREICE*, 2016, vol. 3, Special Issue 1, pp. 109–113.
13. Mourchid Y., Hassouni M. E., Cherif H. A. New Image Segmentation Approach using Community Detection Algorithms, *International Journal of Computer Information Systems and Industrial Management Applications*, 2016, vol. 8, pp. 195–204.
14. Browet A., Absil P.-A., Van Dooren P. Community Detection for Hierarchical Image Segmentation, *IWCIA*, 2011, LNCS 6636, pp. 358–371.
15. Newman M. E. Analysis of weighted networks, *Physical Review E*, 2004, vol. 70, no. 5, pp. 056131.
16. Clauset A., Newman M. E. J., Moore C. Finding community structure in very large networks, *Physical Review E*, 2004, vol. 70, no. 6, pp. 066111.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор Е. В. Конова.

Корректор Е. В. Комиссарова.

Сдано в набор 07.02.2017. Подписано в печать 24.03.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ417. 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.