

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24  
2018  
№ 6

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Гурарий М. М., Русаков С. Г., Ульянов С. Л., Жаров М. М. Итерационный метод решения параметризованной системы линейных алгебраических уравнений с многими правыми частями . . . . . 363
- Агасиев Т. А., Карпенко А. П. Современные техники глобальной оптимизации. Обзор . . . . . 370
- Бронштейн Е. М., Гиндуллин Р. В. Решение задачи о заборе и доставке грузов одним транспортным средством с использованием пакета CPLEX . . . . . 387

### ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

- Асратян Р. Э. Коммуникационно-ориентированная архитектура распределенных информационных систем на основе службы обработки защищенных сообщений . . . . . 393

### НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Абдуллаев Н. Т., Гасанкулиева М. М., Джабиева А. Д. Применение нейросетевой технологии при фликкер-шумовой спектроскопии электрокардиограммы . . . . . 402
- Пантюхин Д. В., Карелова Е. Повышение качества классификации компьютерных атак сверточной нейронной сетью посредством балансировки обучающей выборки . . . . . 406

### ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Макаров С. Л. Информационные технологии поиска аналогов исследования по каталогам диссертаций . . . . . 414
- Шмелева А. Г., Ладынин А. И., Бахметьев А. В. Построение взвешенных решений управления сложными производственными системами с применением теории массового обслуживания . . . . . 421

### ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Назаров Л. Е., Шишкин П. В. Характеристики помехоустойчивых блоковых турбокодов на основе низкоплотностных кодов . . . . . 427

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

- Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M.** The Iterative Method for Solving Parameterized System of Linear Equations with Multiple Right Hand Sides ..... 363
- Agasiev T. A., Karpenko A. P.** Modern Techniques of Global Optimization. Review ..... 370
- Bronstein E. M., Gindullin R. V.** Solving Single Vehicle Routing Problem with Pick-Ups and Deliveries with CPLEX ..... 387

#### INFORMATION SECURITY

- Asratian R. E.** Communication-Oriented Architecture of Distributed Information Systems on the Base of Protected Message Service ..... 393

#### NEUROTECHNOLOGIES

- Abdullaev N. T., Qasankuliyeva M. M., Dzhabiya A. Dj.** Application Of Neuroet Network Technology For Fliker-Noise Spectroscopy Of Electrocardiogram ... 402
- Pantiukhin D. V., Karelova E.** Quality Improvement of Intrusion Classification through Convolution Neural Network by Training on Balanced Samples ..... 406

#### INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Makarov S. L.** Information Technologies of a Search for Similar Researches in Dissertations Catalogues ..... 414
- Shmeleva A. G., Ladynin A. I., Bakhmetiev A. V.** Weighted Decisions Development for Complex Production Systems Management Using the Theory of Mass Service ..... 421

#### DIGITAL PROCESSING OF SIGNALS AND IMAGES

- Nazarov L. E., Shishkin P. V.** The characteristics of Error-Correcting Block Turbo-Codes Based on Low-Density Parity-Check Codes ..... 427

#### 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"

**М. М. Гуарий**, канд. техн. наук, e-mail: gourary@ippm.ru,  
**С. Г. Русаков**, д-р техн. наук, проф., e-mail: rusakov@ippm.ru,  
**С. Л. Ульянов**, д-р техн. наук, e-mail: ulyas@ippm.ru,  
**М. М. Жаров**, канд. техн. наук, e-mail: zarov@ippm.ru,

ФГБУН Институт проблем проектирования в микроэлектронике РАН, Москва

## Итерационный метод решения параметризованной системы линейных алгебраических уравнений с многими правыми частями

*Рассматривается задача решения систем линейных алгебраических уравнений со многими правыми частями и матрицей, зависящей от параметра. Предлагается итерационный метод решения с повторным использованием результатов матрично-векторного умножения, который обеспечивает сокращение вычислительных затрат при решении таких систем. Метод обобщает ряд известных алгоритмов на базе подпространства Крылова. Применение метода показано на примере выполнения периодического малосигнального анализа нелинейных электронных схем.*

**Ключевые слова:** системы линейных алгебраических уравнений, линейные преобразования, ортогонализация, итерационные методы, подпространство Крылова, приближенное решение

### Введение

В ряде задач из различных технических приложений возникает необходимость многократного решения систем линейных алгебраических уравнений (СЛАУ) с различными правыми частями. В данной работе рассматривается задача решения высокоразмерной параметризованной СЛАУ вида

$$A(s_m)x^{(m)} = b^{(m)}, \quad m = 1, \dots, M, \quad (1)$$

где матрица системы линейно зависит от некоторого скалярного параметра  $s$ :

$$A(s) = A' + sA''. \quad (2)$$

Такая задача возникает, например, при расчетах малосигнальных частотных характеристик нелинейных электронных схем ( $s$  — частота) или при решении задач гомотопии, где  $s$  — параметр продолжения. Предполагается, что система (1) имеет высокую размерность.

Решение линейной системы (1) может быть выполнено с помощью как прямых, так и итерационных методов, однако итерационные методы более предпочтительны вследствие высокой размерности системы. Применение итерацион-

ных методов для решения линейных систем с постоянными матрицами высокой размерности позволяет значительно ускорить решение системы и в ряде случаев не имеет альтернативы [1]. Наибольшее распространение получили методы подпространства Крылова QMR [2] и GMRES [3], причем метод GMRES оказался более эффективным при решении практических задач [4]. Однако непосредственное применение метода GMRES для решения линейной системы (1) с многими правыми частями и матрицей (2) становится неэффективным вследствие линейного роста вычислительных затрат с увеличением числа правых частей. Рост затрат обусловлен тем, что в методе используется базис подпространства Крылова, который жестко связан с вектором начальной невязки и значениями элементов матрицы. Поэтому базис, полученный для одного значения параметра  $s$ , не может быть использован для другого значения ввиду изменения элементов матрицы.

Для решения линейных систем с многими правыми частями разработаны несколько итерационных методов [5–9]. В работах [10–13] предложены методы решения линейной системы вида (1) для случая, когда матрица  $A''$  равна

единичной матрице. Для построения итерационного метода в этом случае используется свойство инвариантности подпространства Крылова  $\mathbf{K}_k(A + sI, b) = \mathbf{K}_k(A, b)$ , которое позволяет использовать единый базис подпространства для решения систем с различными значениями параметра  $s$ . Если матрица  $A''$  отлична от единичной, то необходимый вид линейной системы можно было бы получить, умножая ее на обратную к  $A''$  матрицу (предобуславливание). Этот подход требует явного формирования и вычисления обратной матрицы. Так как основной операцией в методах рассматриваемого класса является матрично-векторное умножение, которое может выполняться без явного формирования матриц, то получение обратной матрицы практически исключено. Поэтому актуальной задачей является конструирование алгоритмов решения параметризованных линейных систем вида (1) с произвольными несимметричными матрицами  $A', A''$ .

В работе предлагается подход к последовательному решению линейных систем (1), при котором каждое последующее решение определяется с помощью минимизации невязки при использовании результатов матрично-векторного умножения, вычисленных при получении предыдущих решений. При этом достигается сокращение вычислительных затрат за счет уменьшения операций матрично-векторного умножения и повторного использования найденных векторов. Полученные таким образом базисные векторы определяют приближенное решение линейной системы, но не принадлежат подпространству Крылова. Поэтому вначале (разд. 1) рассмотрены методы минимальной невязки решения СЛАУ, которые допускают генерацию векторов с помощью операций матрично-векторного умножения. Эти методы являются обобщением известных алгоритмов на базе подпространств Крылова. В разд. 2 показано, что дополнительное ускорение предложенных методов минимальной невязки может быть достигнуто за счет порядка и способов применения линейных преобразований. Затем в разд. 3 рассмотрены методы для решения параметризованной линейной системы с многими правыми частями. Представлен алгоритм с использованием ортогонализации Грама — Шмидта и алгоритм без ортогонализации, в котором формирование и разложение матрицы Грама выполняются с учетом элементов матрицы, полученных при решении предыдущих систем. Применение и характеристики предложенных алгоритмов рассматриваются в разд. 4.

## 1. Алгоритм минимальной невязки

Рассматривается приближенное решение линейной системы

$$Ax = b, \quad (3)$$

где  $A$  — квадратная невырожденная матрица.

Пусть имеется вычислительная процедура  $\text{GET}(u_k, v_k)$ , которая генерирует последовательность пар векторов  $u_k, v_k$  таких, что

$$v_k = Au_k, \quad k = 1, 2, \dots \quad (4)$$

Векторы  $u_k$  назовем исходными, а векторы  $v_k$  — результирующими.

Тогда под алгоритмом минимальной невязки на основе матрично-векторного произведения (МНМВП) понимается алгоритм, который находит приближенное решение линейной системы (3) в виде

$$x_K = Uc = \sum_{k=1}^K c_k u_k. \quad (5)$$

Здесь  $U$  — матрица, столбцами которой служат векторы  $u_k$ ;  $c$  — вектор-столбец коэффициентов  $c_k$ , которые находятся из условия минимума нормы невязки линейной системы (3):

$$\|Ax_K - b\|_2 \rightarrow \min. \quad (6)$$

Принимая во внимание (4—6), можно записать

$$\|r_K\| = \left\| \sum_{k=1}^K c_k v_k - b \right\| \rightarrow \min. \quad (7)$$

Таким образом, для определения решения в виде (5) необходимо решить задачу минимизации (7) относительно коэффициентов  $c_k$ . Задача может быть решена с помощью формирования ортонормированного базиса в пространстве результирующих векторов  $\hat{v}_1, \dots, \hat{v}_K$  и соответствующих им исходных векторов  $\hat{u}_k$ ,  $\hat{v}_k = A\hat{u}_k$ . Тогда коэффициенты приближенного решения

$$x_K = \hat{U}\hat{c} = \sum_{k=1}^K \hat{c}_k \hat{u}_k \quad (8)$$

определяются с помощью проекции результирующих векторов на вектор  $b$ :

$$\hat{c}_k = (\hat{v}_k, b). \quad (9)$$

Векторы  $\hat{v}_k$  могут быть получены с помощью процесса ортогонализации Грама — Шмидта [1]:

$$\hat{v}_k = v_k - \sum_{i=1}^k (v_k, \hat{v}_i) \hat{v}_i, \quad \hat{v}_k = \hat{v}_k / \|\hat{v}_k\|. \quad (10)$$

Исходные векторы определяются аналогично:

$$\hat{u}_k = u_k - \sum_{i=1}^k (v_k, \hat{v}_i) \hat{u}_i, \hat{u}_k = \hat{u}_k / \|v_k\|. \quad (11)$$

Справедливы также рекуррентные соотношения

$$r_k = r_{k-1} - \hat{c}_k \hat{v}_k, r_0 = b; \quad (12)$$

$$x_k = x_{k-1} + \hat{c}_k \hat{u}_k, x_0 = 0. \quad (13)$$

Процесс вычислений по формулам (9)—(13) может быть записан в виде следующего алгоритма МНМВП.

*Шаг 1.* Инициализация  $r = b, x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k - 1$

$v_k = v_k - (v_j, v_k)v_j, u_k = u_k - (v_j, v_k)u_j$

Шаг 2.3.  $v_k = v_k / \|v_k\|, u_k = u_k / \|v_k\|,$

Шаг 2.4.  $r = r - (v_k, r)v_k, x = x + (v_k, r)u_k$

В процедуре GET( $u_k, v_k$ ) выполняется назначение исходного вектора  $u_k$  и вычисление результирующего вектора  $v_k$ . В общем случае исходный вектор  $u_k$  рассчитывается по формуле

$$u_k = P^{-1}(\alpha_k \hat{v}_{k-1} + \beta_k r_{k-1}), v_k = Au_k, \quad (14)$$

где  $\alpha_k, \beta_k$  — коэффициенты и  $\alpha_1 = 0, P$  — матрица правого преобуславливателя.

В зависимости от выбора коэффициентов исходного вектора  $u_k$  в процедуре Get( $u_k, v_k$ ) можно получить те или иные известные алгоритмы. Например, выбор  $\alpha_k = 0, \beta_k = 1$  соответствует алгоритму GCR [1], а  $\alpha_k = 1, \beta_k = 0$  дает

$$u_k = P_k^{-1} \hat{v}_{k-1}, v_k = Au_k, \quad (15)$$

что приводит к алгоритму ORTHODIR [1].

## 2. Сокращение вычислительных затрат в алгоритме МНМВП

1. **Алгоритм МНМВП-Т.** Линейное преобразование исходных векторов (11) требует значительных вычислительных затрат. Этих затрат можно избежать, если применить это линейное преобразование не к исходным векторам, а к вектору коэффициентов  $\hat{c}$  (9) для получения вектора  $c$ . Для получения необходимого преобразования отметим, что процесс ортого-

нализации (10) может быть записан в форме QR разложения матрицы результирующих векторов  $V$ :

$$V = \hat{V}\Gamma, \quad (16)$$

где  $\Gamma$  — верхняя треугольная матрица с элементами  $\gamma_{ij} = (v_j, \hat{v}_i)$ .

Аналогично для матрицы исходных векторов

$$U = \hat{U}\Gamma. \quad (17)$$

Поэтому вектор решения (5) имеет вид

$$x_K = Uc = \hat{U}\Gamma c = \hat{U}(\Gamma c), \quad (18)$$

откуда с учетом (8) получаем

$$c = \Gamma^{-1} \hat{c}. \quad (19)$$

Тогда алгоритм МНМВП можно переписать, включив формирование и решение системы с верхней треугольной матрицей.

**Алгоритм МНМВП-Г**

*Шаг 1.* Инициализация  $r = b, x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k - 1$

$\gamma_{jk} = (v_j, v_k), v_k = v_k - \gamma_{jk}v_j$

Шаг 2.3.  $\gamma_{kk} = \|v_k\|, v_k = v_k / \gamma_{kk}$

Шаг 2.4.  $c_k = (v_k, r), r = r - c_k v_k$

*Шаг 3.*  $c = \Gamma^{-1}c, x = \sum_{i=1}^k c_i u_i$

Здесь вектор  $c$  содержит элементы вектора  $\hat{c}$  и на последнем шаге элементы вектора  $c$ .

2. **Алгоритм МНМВП-Г.** Представим норму невязки (7) в виде квадратной функции аргумента  $c_k$ :

$$\|r_k\|^2 = c^T Gc - 2p^T c + \|b\|^2 \rightarrow \min. \quad (20)$$

Здесь  $G = V^T V$  — матрица Грама с элементами  $g_{ij} = (v_i, v_j), p$  — вектор с компонентами  $p_i = (b, v_i)$ . Решение линейной системы

$$Gc = p \quad (21)$$

доставляет минимум выражения (20).

Поскольку матрица Грама симметричная и положительно определенная, то для решения системы (21) целесообразно использовать разложение Холецкого, т.е. представление матрицы в виде произведения транспонированной верхней треугольной матрицы на верхнюю треугольную матрицу.

Для матрицы Грама верхняя треугольная матрица разложения Холецкого совпадает с верхней треугольной матрицей  $\Gamma$  (16):

$$G = V^T V = (\hat{V}\Gamma)^T (\hat{V}\Gamma) = \Gamma^T (\hat{V}^T \hat{V}) \Gamma = \Gamma^T \Gamma. \quad (22)$$

Поэтому для определения коэффициентов решения необходимо решить две линейные системы с нижней и верхней треугольными матрицами:

$$\hat{c} = \Gamma^{-T} p; \quad (23)$$

$$c = \Gamma^{-1} \hat{c}. \quad (24)$$

Текущий вектор невязки в этом случае определяется по формуле

$$r_k = b - \sum_{i=1}^k c_k v_k \quad (25)$$

или по рекуррентному соотношению

$$r_k = r_{k-1} - \hat{c}_k \left( \sum_{i=1}^{k-1} \gamma_{ik} v_i / \gamma_{kk} + v_k \right). \quad (26)$$

Тогда алгоритм МНМВП, включающий разложение матрицы Грама и решение систем с треугольной матрицей (алгоритм МНМВП-Г), имеет следующий вид:

*Шаг 1.* Инициализация  $r = b$ ,  $x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k$

Если  $j < k$

$$\gamma_{jk} = \left( (v_j, v_k) - \sum_{i=1}^{j-1} \gamma_{ik} \gamma_{ij} \right) / \gamma_{jj}$$

Иначе  $\gamma_{kk} = \sqrt{(v_k, v_k) - \sum_{i=1}^{k-1} \gamma_{ik}^2}$

Шаг 2.3.  $c_k = \left( (b, v_k) - \sum_{i=1}^{k-1} \gamma_{ik} c_i \right) / \gamma_{kk}$

Шаг 2.4  $r = r - c_k \left( \sum_{i=1}^{k-1} \gamma_{ik} v_i / \gamma_{kk} + v_k \right)$

*Шаг 3.*  $c = \Gamma^{-1} c$ ,  $x = \sum_{i=1}^k c_i u_i$

На шаге 2.2 выполняется разложение Холецкого матрицы Грама. На шаге 2.3 находится решение линейной системы с нижней треугольной матрицей (23).

### 3. Алгоритмы решения параметризованной линейной системы

1. Алгоритм ММН с ортогонализацией. Нашей целью является построение такого алго-

ритма решения линейных систем (1), в котором для получения последующего решения используются результаты матрично-векторного умножения, вычисленные при получении предыдущих решений. Новый алгоритм получается на основе алгоритма МНМВП-Т.

Матрично-векторное произведение с матрицей (2) имеет вид

$$v(s) = A(s)u = v' + sv'', \quad (27)$$

где

$$v' = A'u; \quad v'' = A''u. \quad (28)$$

Из выражения (27) следует, что если вычислить векторы  $v'$ ,  $v''$  при некотором значении параметра  $s = s_1$ , то матрично-векторное произведение  $A(s)u$  легко может быть получено для любого другого значения параметра по формуле (27). Такая операция включает умножение вектора на скаляр и сложение двух векторов. При этом вычислительные затраты значительно меньше затрат на выполнение нового матрично-векторного умножения. Вычисленные векторы необходимо хранить в памяти программы. При добавлении нового вектора произведения для получения ортонормированного базиса выполняется ортогонализация. Так как найденные базисные векторы уже не принадлежат пространству Крылова, то новый вектор произведения может оказаться линейной комбинацией векторов текущего базиса. В этом случае происходит обрыв вычислительного процесса. Проверка условия

$$(v_k, r_k) \neq 0 \quad (29)$$

позволяет определить возможность обрыва.

При выполнении этого условия обрыв процесса исключается, так как текущий вектор невязки ортогонален базисным векторам. Если условие (29) не выполняется, возможно выполнение одного из следующих действий:

— пропуск вектора, если в условии (29) участвует вектор произведения из памяти программы;

— перевычисление вектора, полученного с помощью матрично-векторного умножения, по формуле

$$v_k = (AP^{-1})^m v_k, \quad (30)$$

где  $m$  — целое, при котором удовлетворяется условие (29).

Тогда процедура получения пары векторов Get  $(u_k, v_k)$  имеет вид

Шаг 1.  $l_k = l_k + 1$

Шаг 2. Если  $l_k > n$

Шаг 2.1.  $n = l_k$

Шаг 2.2. Если  $breakdown = NO$

$u_n = P^{-1}r$

иначе  $u_n = P^{-1}v_k$

Шаг 2.3.  $v'_n = A'u_n$ ;  $v''_n = A''u_n$

Шаг 3.  $v_k = v'_{l_k} + s_m v''_{l_k}$

Шаг 4. Если  $(v_k, r) = 0$

$breakdown = YES$ ;

Переход на Шаг 1.

Здесь  $n$  — число пар векторов  $u_k, v_k$ , хранимых в памяти программы;  $l_k$  — индекс пары векторов на шаге  $k$  алгоритма МНМВП-Т.

Отметим, что шаг инициализации алгоритма теперь имеет вид  $r = b^{(m)}$ ,  $x^{(m)} = 0$ , шаг вычисления приближенного решения —  $x^{(m)} = \sum_{i=1}^k c_{l_i} u_i$ . Остальные шаги алгоритма МНМВП-Т

не изменяются. Вариант алгоритма МНМВП-Т для решения параметризованных линейных систем назовем ММН алгоритмом.

## 2. Алгоритм ММН без ортогонализации.

ММН алгоритм без ортогонализации может быть построен на базе алгоритма МНМВП-Г, в котором исключаются вычислительные затраты на ортогонализацию векторов. Кроме того, можно снизить затраты на получение матрицы Грама.

Используя определение матрицы Грама, можно записать ее элементы:

$$g_{ij}(s) = (v_i(s), v_j(s)) = g'_{ij} + s g''_{ij} + s^2 g'''_{ij}, \quad (31)$$

где

$$g'_{ij} = (v'_i, v'_j); \quad g''_{ij} = (v'_i, v''_j) + (v''_i, v'_j); \quad g'''_{ij} = (v''_i, v''_j). \quad (32)$$

Из выражений (31, 32) следует, что, располагая значениями четырех скалярных произведений, можно вычислить элементы матрицы для любого параметра  $s$ . Кроме того, получив элементы матрицы для трех значений параметра  $s^{(1)}, s^{(2)}, s^{(3)}$ , можно использовать интерполяционную формулу Лангранжа [14]:

$$g_{ij}(s) = L(s, g_{ij}^{(1)}, g_{ij}^{(2)}, g_{ij}^{(3)}, s^{(1)}, s^{(2)}, s^{(3)}) = \\ = \frac{g_{ij}^{(1)}(s - s^{(2)})(s - s^{(3)})}{(s^{(1)} - s^{(2)})(s^{(1)} - s^{(3)})} + \frac{g_{ij}^{(2)}(s - s^{(1)})(s - s^{(3)})}{(s^{(2)} - s^{(1)})(s^{(2)} - s^{(3)})} + \\ + \frac{g_{ij}^{(3)}(s - s^{(1)})(s - s^{(2)})}{(s^{(3)} - s^{(2)})(s^{(3)} - s^{(1)})}, \quad (33)$$

где  $g_{ij}^{(l)} = g_{ij}(s^{(l)})$ .

Дополнительно можно исключить вычисление на каждой итерации вектора невязки по формуле (25), а для расчета нормы вектора невязки для проверки сходимости алгоритма использовать следующее выражение:

$$\|r_k\|^2 = \|r_{k-1}\|^2 - \hat{c}_k^2. \quad (34)$$

Ниже представлен алгоритм, использующий формулы (33), (34) для решения системы линейных уравнений (1) с параметром  $s_m$ .

### Алгоритм ММН-Г.

Шаг 1. Инициализация  $r = b^{(m)}$ ,  $\rho = \|r\|$ ,  $x = 0$

Шаг 2. Цикл  $k = 1, 2, \dots$

Шаг 2.1. Если  $k > n$

Шаг 2.1.1.  $n = k$ ;  $l_k = 0$

Шаг 2.1.2.  $u_k = P^{-1}r$ ;  $v'_k = A'u_k$ ;  $v''_k = A''u_k$

Шаг 2.2.  $v_k = v'_k + s_m v''_k$

Шаг 2.3.  $l_k = l_k + 1$ ,  $l = l_k$

Шаг 2.4. Если  $l < 3$

$s_k^{(l)} = s_m$

Шаг 2.5. Цикл  $j = 1, \dots, k$

Шаг 2.5.1.

Если  $(l > 3)$

$g = L(s_m, g_{kj}^{(1)}, g_{kj}^{(2)}, g_{kj}^{(3)}, s_k^{(1)}, s_k^{(2)}, s_k^{(3)})$

Иначе

$g_{kj}^{(l)} = (v_k, v_j)$ ,  $g = g_{kj}^{(l)}$

Шаг 2.5.2.

Если  $j < k$

$\gamma_{jk} = \left( g - \sum_{i=1}^{j-1} \gamma_{ik} \gamma_{ij} \right) / \gamma_{jj}$

Иначе

$\gamma_{kk} = \sqrt{g - \sum_{i=1}^{k-1} \gamma_{ik}^2}$

Шаг 2.6.  $c_k = \left( (b^{(m)}, v_k) - \sum_{i=1}^{k-1} \gamma_{ik} c_i \right) / \gamma_{kk}$

Шаг 2.7.  $\rho^2 = \rho^2 - c_k^2$

Шаг 2.8. Если  $k > n$

$r = r - \frac{c_k}{\gamma_{kk}} \sum_{i=1}^{k-1} \gamma_{ik} v_i - c_k v_k$

Шаг 2.9. Если  $k = n$

$d = \Gamma^{-1}c$ ,  $r = r - \sum_{i=1}^k d_i v_i$

Шаг 3.  $c = \Gamma^{-1}c$ ,  $x^{(m)} = \sum_{i=1}^k c_i u_i$ .

Здесь переменные  $s_k^{(l)}$  задают значения параметра  $s$ , для которых определяются элементы матриц  $g_{kj}^{(1)}, g_{kj}^{(2)}, g_{kj}^{(3)}$ . Так как матрица Грама симметричная, то в памяти программы достаточно хранить только верхнюю треугольную часть матрицы.

#### 4. Численный эксперимент

Работоспособность и эффективность предложенных алгоритмов иллюстрируется на примере их использования в периодическом малосигнальном анализе. Типичным применением этого анализа является моделирование схемы смесителя, на входы которой подаются два сигнала разной частоты  $\Omega$ ,  $\omega$ . Один из этих сигналов считается большим, а второй — малым. В отличие от традиционного малосигнального анализа в периодическом малосигнальном анализе вначале определяется периодический установившийся режим по большому сигналу, а затем рассчитывается реакция схемы на малое синусоидальное воздействие, частота которого должна варьироваться в заданном диапазоне. В периодическом малосигнальном анализе на основе гармонического баланса [15] периодический установившийся режим задается значениями гармоник частоты большого сигнала с индексами  $k = -K_h, \dots, K_h$ , а линейная система решается относительно комбинационных частот с индексами  $[k, +1]$ , где 1 соответствует первой гармонике малого сигнала. Линейная система имеет вид [15]

$$A(\omega)x(\omega) = b(\omega), \quad (35)$$

где  $\omega$  — частота малого сигнала. Размерность линейной системы  $(2K_h + 1)N$ , где  $N$  — число узловых переменных схемы.

Периодический малосигнальный анализ позволяет рассчитать зависимости комбинационных компонент отклика схемы  $|\omega + k\Omega|$  от частоты входного малосигнального воздействия  $\omega$ . Ниже приведены результаты анализа для трех схем, характеристики которых представлены в табл. 1.

Схема 1 — это одностранзисторный смеситель, который содержит восемь элементов (один транзистор, два резистора, два конденсатора, одна индуктивность и два источника напряжения). На схему подаются два входных сигнала с частотой 1 МГц, амплитудой 0,03 В и с частотой 95...105 кГц, амплитудой 0,001 В. Преобразователь частоты (схема 2) содержит

Таблица 1

Схема	Число элементов	$N$	$K_h$	$(2K_h + 1)N$
1	8	11	30	671
2	19	16	30 60	976 1936
3	99	59	10 30 60	1239 3599 7139

19 элементов. Входные сигналы имеют частоту 140 МГц с амплитудой 0,5 В и амплитудой 0,0125 В с частотой из диапазона 9,5...10,5 МГц. Схема 3 — смеситель на основе ячейки Гильберта, последовательно с которым включены фильтр и широкополосный усилитель. Эта схема содержит 17 транзисторов, 47 резисторов, 30 конденсаторов и 5 индуктивностей. Входной сигнал имеет частоту 1 ГГц и амплитуду 0,4 В, частота малосигнального воздействия изменяется в диапазоне 860...900 МГц.

Вычислительные затраты периодического малосигнального анализа определяются затратами на решение систем линейных уравнений (35). В итерационных методах рассматриваемого класса эти затраты в основном зависят от затрат на матрично-векторное умножение и операцию ортогонализации векторов. В табл. 2 приведено количество матрично-векторных умножений для метода GMRES, ММН, ММН-Г (3—4 столбцы). Как следует из этой таблицы (столбцы 6, 7), предложенные алгоритмы решения позволяют снизить общее число матрично-векторных умножений в 30—40 (ММН) или в 12—35 (ММН-Г) раз.

Вместе с тем суммарные затраты на построение ортонормированного базиса в алгоритме ММН выше, чем в алгоритме GMRES, так как число векторов в алгоритме ММН больше, чем их число в алгоритме GMRES в пересчете на одну частотную точку (столбцы 3, 4 в табл. 3). Вследствие этого суммарный вычислительный выигрыш определяется соотношением трудоемкости операций матрично-векторного умножения и ортогонализации. Для схемы 3 суммарный выигрыш показан в табл. 3 (столбец 6).

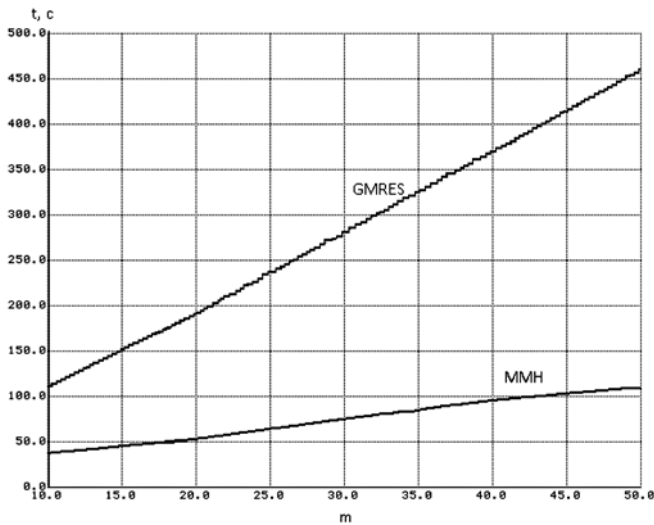
Таблица 2

Схема	$K_h$	#mv_gmres	#mv_ммн	#mv_ммн-г	#mv_gmres / #mv_ммн	#mv_gmres / #mv_ммн-г
1	30	510	18	41	28	12
2	30 60	612 625	19 21	31 32	32 29,8	19,7 19,5
3	10 60	1836 1836	42 45	48 52	43,7 40,8	38 35



Таблица 3

Схема	$K_n$	$t_{\text{ort\_gmres}}$ , с	$t_{\text{ort\_ммн}}$ , с	$t_{\text{ort\_ммн}}/$ $t_{\text{ort\_gmres}}$	$t_{\text{gmres}}/$ $t_{\text{ммн}}$
3	10	5,1	16	3,1	4,1
	60	15	41	2,7	3,5



Зависимость вычислительных затрат в алгоритме GMRES и MMH от числа значений частоты малосигнального возбуждения

В алгоритме MMH-Г затраты на построение ортонормированного базиса отсутствуют, поэтому вычислительный выигрыш здесь может быть выше. В частности, для схемы 3 выигрыш составляет 5—6 раз. Однако, как показали эксперименты, при увеличении амплитуды большого сигнала может наблюдаться расходимость метода.

Число операций матрично-векторного умножения, вычислительные затраты на их выполнение и ортогонализацию векторов зависят от числа точек изменения частоты малосигнального возбуждения, а также от ширины диапазона ее изменения. На рисунке представлены вычислительные затраты в зависимости от числа значений частоты малого сигнала.

### Заключение

В работе рассмотрены итерационные методы решения параметризованной системы линейных алгебраических уравнений с многими правыми частями. Предложены алгоритмы, построенные на использовании результатов матрично-векторного умножения и повторного использования найденных векторов. Показано, что эти алгоритмы применимы для решения линейных систем с матрицами, коэффициенты которых линейно зависят от скалярного параметра.

Предложенные алгоритмы используются для получения эффективной вычислительной процедуры периодического малосигнального анализа на основе гармонического баланса. Проведенные численные эксперименты по моделированию нелинейных электронных схем показали, что вычислительные затраты определяются операциями матрично-векторного умножения и построения ортонормированного базиса векторов. Предложенный алгоритм MMH с ортогонализацией позволяет значительно снижать число операций умножения при некотором росте затрат на ортогонализацию. Алгоритм MMH без ортогонализации позволяет получить больший вычислительный выигрыш, но его применимость ограничена вследствие возможной расходимости алгоритма.

### Список литературы

1. Saad Y. Iterative methods for Sparse Linear Systems. PWS publishing company, 1996.
2. Freund R. W. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems // SIAM J. Sci. Stat. Comput. 1993. Vol. 14, N. 2. P. 470—482.
3. Saad Y., Schultz M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems // SIAM J. Sci. and Statist. Comput. 1986. Vol. 7, N. 3. P. 856—869.
4. Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M. et al. Iterative Solution of Linear Systems in Harmonic Balance Analysis // IEEE MTT-S Int. Microwave Symposium Digest. 1997. P. 1507—1510.
5. Boyse W. E., Seidel A. A. A block QMR method for computing multiple simultaneous solutions to complex symmetric systems // SIAM J. Sci. Comput. 1996. Vol. 17, N. 1. P. 263—274.
6. Freund R. and Malhotra M. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides // Linear Alg. Appl. 1997. Vol. 254, N. 1—3. P. 197—257.
7. Simoncini V., Gallopoulos E. An iterative method for nonsymmetric systems with multiple right-hand sides // SIAM J. Sci. Comput., 1995. Vol. 16, N. 4. P. 917—933.
8. Simoncini V., Gallopoulos E. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides // J. Comput. Appl. Math. 1996. Vol. 66, N. 1. P. 457—469.
9. Baker A. H., Dennis J. M., Jessup E. R. On improving linear solver performance: A block variant of GMRES // SIAM J. Sci. Comput. 2006. Vol. 27, N. 5. P. 1608—1626.
10. Simoncini V. Restarted full orthogonalization method for shifted linear systems // BIT Numer. Math. 2003. Vol. 43, N. 2. P. 459—466.
11. Gu G-D., Simoncini V. Numerical solution of parameter-dependent linear systems // Linear Alg. Appl. 2005. Vol. 12, N. 9. P. 923—940.
12. Baumann M., Gijzen M. B. Nested Krylov methods for shifted linear systems // SIAM J Sci Comput. 2015. Vol. 37, N. 5. P. S90—S112.
13. Darnell D., Morgan R. B., Wilcox W. Deflated GMRES for systems with multiple shifts and multiple right-hand sides // Linear Algebra Appl. 2008. Vol. 429, N. 10. P. 2415—2434.
14. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). М.: Высшая школа, 2001.
15. Гурарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л. Метод малосигнального анализа для моделирования многочастотных радиотехнических схем // В сб. трудов Всероссийской научно-технической конференции "Проблема разработки перспективных микроэлектронных систем — 2008" (МЭС-2008). М.: ИППМ РАН, 2008. С. 71—76.

## The Iterative Method for Solving Parameterized System of Linear Equations with Multiple Right Hand Sides

*The problem of solving a system of linear equations with multiple right hand sides and matrix dependent on the parameter is considered. The iterative method with reuse of the previous results of matrix vector multiplications is suggested. The method provides the reduction of computational efforts and decrease of matrix-vector multiplications due to reuse of computed vectors. The proposed method is a generalization of known Krylov subspace algorithms. The application of proposed algorithms for periodic small-signal analysis of nonlinear electronic circuits is shown.*

**Keywords:** system of linear algebraic equations, linear transform, orthogonalization, iterative methods, Krylov subspace, approximate solution

### References

1. Saad Y. *Iterative methods for Sparse Linear Systems*. Boston, PWS publishing company, 1996.
2. Freund R. W. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Stat. Comput.* 1993, vol. 14, no. 2, pp. 470–482.
3. Saad Y., Schultz M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. and Statist. Comput.*, 1986, vol. 7, no. 3, pp. 856–869.
4. Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M. et al. Iterative Solution of Linear Systems in Harmonic Balance Analysis, *IEEE MTT-S Int. Microwave Symposium Digest*, 1997, pp. 1507–1510.
5. Boyse W. E., Seidel A. A. A block QMR method for computing multiple simultaneous solutions to complex symmetric systems, *SIAM J. Sci. Comput.*, 1996, vol. 17, no. 1, pp. 263–274.
6. Freund R., Malhotra M. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, *Linear Alg. Appl.*, 1997, vol. 254, no. 1–3, pp.197–257.
7. Simoncini V., Gallopoulos E. An iterative method for non-symmetric systems with multiple right-hand sides, *SIAM J. Sci. Comput.*, 1995, vol. 16, no. 4, pp. 917–933.
8. Simoncini V., Gallopoulos E. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides, *J. Comput. Appl. Math.*, 1996, vol. 66, no. 1, pp. 457–469.
9. Baker A. H., Dennis J. M., Jessup E. R. On improving linear solver performance: A block variant of GMRES, *SIAM J. Sci. Comput.*, 2006, vol. 27, no. 5, pp. 1608–1626.
10. Simoncini V. Restarted full orthogonalization method for shifted linear systems, *BIT Numer. Math.*, 2003, vol. 43, no. 2, pp. 459–466.
11. Gu G-D., Simoncini V. Numerical solution of parameter-dependent linear systems, *Linear Alg. Appl.*, 2005, vol. 12, no. 9, pp. 923–940.
12. Baumann M., Gijzen M. B. Nested Krylov methods for shifted linear systems, *SIAM J Sci Comput.*, 2015, vol. 37, no. 5, pp. S90–S112.
13. Darnell D., Morgan R. B., Wilcox W. Deflated GMRES for systems with multiple shifts and multiple right-hand sides, *Linear Algebra Appl.*, 2008, vol. 429, no. 10, pp. 2415–2434.
14. Verbitsky V. M. *Chislennyye metody (mathematischeskiy analiz i jbknoventnye differentsialnye uravneniya)*, Moscow: Vysshaya shkola, 2001 (in Russian).
15. Gourary M. M., Zharov M. M., Rusakov S. G., Ulyanov S. L. Metod malosignalnogo analiza dla modelirovaniya mnogochastotnykh radiotekhnicheskikh shem, *Sb. trudov Vserossiyskoy nauchno-tekhnicheskoy konferentsii "Problemy razrabotki perspektivnykh mikroelektronnykh shem"*. Moscow: IPPM RAN, 2008, pp. 71–76 (in Russian).

УДК 519.6

DOI: 10.17587/it.24.370-386

T. A. Агасиев, аспирант, e-mail: agtaleh@mail.ru,  
А. П. Карпенко, д-р физ.-мат. наук, проф., зав. кафедрой, e-mail: arkarpenko@mail.ru,  
Московский государственный технический университет им. Н. Э. Баумана

## Современные техники глобальной оптимизации. Обзор

*Представляем постановку задачи глобальной условной оптимизации, основной особенностью которой является высокая вычислительная сложность целевой функции. Даем определения таких сущностей, как характерные признаки задачи, базовые задача и алгоритм оптимизации, метазадача и метаалгоритм оптимизации, стратегия базовой задачи, индикатор эффективности стратегии. Определяем и приводим постановки мультииндикаторной, мультиклассовой и мультибюджетной задач метаоптимизации. На основе анализа около 100 публикаций даем обзор современных методов ландшафтного анализа целевых функций, а также методов метаоптимизации базовых алгоритмов. Наконец, представляем краткий обзор современного программного обеспечения, используемого для решения базовых и метазадач оптимизации.*

**Ключевые слова:** глобальная оптимизация, ландшафтный анализ, метаоптимизация, суррогатное моделирование, анализ эффективности алгоритмов оптимизации

### Введение

Задачи глобальной оптимизации возникают во многих приложениях, например, при раз-

работке комплексных инженерных изделий. Поэтому современные системы автоматизированного проектирования (САД) и инженерного анализа (САЕ) включают в себя программы,

реализующие, в том числе, алгоритмы непрерывной глобальной оптимизации.

Значительная вычислительная сложность моделей оптимизируемых объектов не позволяет осуществлять большое число обращений к этой модели. Поэтому в современном "промышленном" программном обеспечении задач оптимизации широко используют предварительные анализ и обработку этих задач, включая анализ исходных данных, понижение размерности пространства поиска, ландшафтный анализ целевой функции и т.д. [1]. По той же причине широко применяют предварительный выбор (с помощью методов метаоптимизации) наиболее эффективного алгоритма оптимизации, учитывающего особенности исходной оптимизационной задачи. Методологической основой этой техники является известная теорема о бесплатных завтраках (*No Free Lunch theorem* [2]), из которой следует, что если алгоритм  $a_1$  эффективнее алгоритма  $a_2$  при решении некоторой задачи, то обязательно найдется задача, для которой алгоритм  $a_2$  окажется эффективнее алгоритма  $a_1$ .

Для подавляющего большинства практически значимых задач оптимизации неизвестны аналитические выражения целевой и ограничивающих функций. Поэтому общей практикой является применение прямых алгоритмов глобальной оптимизации, не использующих информацию о производных целевой функции [3]. Для достижения высокой эффективности в процессе решения сложных оптимизационных задач используют составные алгоритмы, комбинирующие различные "простые" алгоритмы оптимизации. Широко применяют, прежде всего, стохастические алгоритмы оптимизации: алгоритм симуляции отжига [4]; эволюционные, в том числе генетические алгоритмы [5, 6]; метаэвристические алгоритмы, вдохновленные природой, такие как алгоритм роя частиц [7], роя пчел [8], колонии муравьев [9] и т.д.

Для уменьшения числа обращений к целевой функции (числа *испытаний*) в процессе решения задач оптимизации, имеющих высокую вычислительную сложность, применяют аппроксимирующие модели целевой функции (*метамодели, суррогатные функции*). Примерами методов, использующих эту технику, являются *Efficient global optimization (EGO)* [10], *Sequential design for optimization (SDO)* [11], *SNOBFIT* [12].

Анализ эффективности алгоритмов оптимизации осуществляют путем экспериментально-го тестирования этих алгоритмов. Особенности

проведения вычислительного эксперимента в целях оценки эффективности оптимизационных алгоритмов рассмотрены, например, в работе [13]. Для анализа эффективности алгоритмов оптимизации и их сравнения составлено большое число открытых наборов тестовых и инженерных задач. Значительная часть онлайн наборов таких задач собрана в коллекции Миттлмана (*Benchmarks for Optimization Software By Hans Mittelmann*) [14]. Наиболее полный обзор источников тестовых задач глобальной оптимизации выполнен в статье [15]. Набор *CUTE (Constrained and Unconstrained Testing Environment)* [16] содержит в формате *SIF* несколько сотен задач условной и безусловной оптимизации. Значительное число различных наборов тестовых задач и ссылок на соответствующую литературу собрано в [17]. Сложные инженерные задачи представлены в статье [18] и книге [19].

Широкий обзор существующих подходов к анализу результатов вычислительных экспериментов по исследованию эффективности оптимизационных алгоритмов выполнен в работе [20]. Для визуального сравнения результатов оценки эффективности алгоритмов оптимизации широкое распространение получили *профили производительности (Performance Profiles)* [21]. Авторы работы [22] рекомендуют использовать эту технику для попарного сравнения алгоритмов оптимизации в целях получения оценок их относительной эффективности. В работе [23] предложено использовать *профили данных (Data Profiles)* как дополнительное средство анализа эффективности алгоритмов оптимизации в условиях заданных ограничений на доступные вычислительные ресурсы.

Работа организована следующим образом. В п. 1 представляем постановку задачи и даем основные определения. В пп. 2, 3 рассматриваем две современные техники, используемые при "промышленном" решении задач глобальной оптимизации, — техники ландшафтного анализа целевой функции и метаоптимизации. В п. 3 представляем краткий обзор современного программного обеспечения, используемого для решения исходной задачи оптимизации и задачи метаоптимизации.

## 1. Постановка задачи и основные определения

**Постановка задачи.** Рассматриваем задачу  $q$  глобальной условной оптимизации

$$\min_{X \in D_X} f(X) = f(X^*) = f^*, \quad (1)$$

где  $X = (x_1, x_2, \dots, x_{|X|})$  —  $|X|$ -мерный вектор варьируемых параметров (размерность задачи  $q$ );  $D_X \subset \mathbb{R}^{|X|}$  — область поиска;  $f(X)$  — целевая функция;  $X^*, f^*$  — искомые оптимальный вектор  $X$  и значение целевой функции.

**Характерные признаки задачи.** Результатом предварительной оценки свойств задачи  $q = q(C)$  является  $|C|$ -мерный вектор *характерных признаков* (ХП)  $C = (c_1, c_2, \dots, c_{|C|})$  этой задачи. Различаем априорные и апостериорные ХП.

*Априорные* ХП прямо вытекают из постановки задачи (1) и включают в себя: размерность  $|X|$  пространства варьируемых параметров; признак наличия или отсутствия ограничивающих функций в определении области  $D_X$ ; тип целевой и ограничивающих функций; прогнозируемое число локальных экстремумов и т.д.

Авторы упомянутого выше набора *CUTE* тестовых задач оптимизации предложили универсальную систему обозначений задач оптимизации, основанную на использовании их априорных ХП [24]. Эта система каждой задаче  $q(C)$  ставит в соответствие код вида

$$c_1 c_2 c_3 c_4 - c_5 c_6 - c_7 - c_8.$$

Здесь приняты следующие обозначения:  $c_1$  определяет тип целевой функции (целевая функция не определена, является константной, линейной, квадратичной, представляет собой сумму квадратов, имеет общий вид);  $c_2$  кодирует тип ограничений;  $c_3, c_4$  представляют гладкость и степень дифференцируемости целевой функции;  $c_5$  указывает на источник задачи (тестовая, имитирующая реальную задачу с отсутствием известного решения, практически значимая реальная задача оптимизации);  $c_6$  свидетельствует о наличии или отсутствии дополнительных составных компонентов целевой функции, например вспомогательных функций в ее составе; величины  $c_7 = |X|$ ,  $c_8$  обозначают размерность задачи и число ограничивающих функций соответственно. Данную систему обозначений активно используют при тестировании алгоритмов оптимизации как позволяющую значительно упростить поиск и отбор необходимых тестовых задач.

*Апостериорные* ХП, в отличие от априорных, требуют вычислительных затрат на предварительные испытания целевой функции в области поиска  $D_X$  в целях последующей экспертной и/или автоматической оценки результатов испытаний. Подчеркнем, что полученная при этом информация о результатах испытаний целевой функции может быть в последу-

ющем использована для решения задачи  $q(C)$ . К числу апостериорных ХП задачи оптимизации относят, прежде всего, ХП целевой функции.

Определение значений апостериорных ХП посредством экспертной оценки имеет ряд очевидных недостатков: низкая точность оценки, высокие временные затраты эксперта и т.д. Методы оценки значений апостериорных ХП целевой функции без использования экспертных оценок называют методами *ландшафтного анализа* (ЛА) [25].

**Базовая и метазадачи оптимизации.** Современные алгоритмы оптимизации имеют, как правило, значительное число свободных параметров  $(b_1, \dots, b_{|B|}) = B$ , от значений которых может существенно зависеть эффективность этих алгоритмов. Отсюда возникает задача *настройки алгоритма* — задача определения в каком-то смысле наилучших ("оптимальных") значений этих параметров. Вектор  $B$  свободных параметров алгоритма оптимизации  $a = a(B)$  называют *стратегией алгоритма*  $a$  (*Б-стратегией*). Множество  $D_B$  допустимых Б-стратегий определяет набор алгоритмов  $A(B) = \{a(B), B \in D_B\}$ , т. е. *метод оптимизации*  $A(B)$ .

Выделяют два типа параметров в составе Б-стратегии — *числовые* и *категориальные*. Числовые параметры могут быть как непрерывными, так и дискретными. Категориальные параметры имеют относительно небольшое число допустимых значений, например, топологий соседства в алгоритме роя частиц [7].

Обычно при решении задачи  $q(C)$  используют "умолчательные" стратегии алгоритма  $a(B)$ , полученные в результате проведенной ранее настройки этого алгоритма его авторами. Настройку алгоритма  $a = a(B)$  может осуществлять также пользователь, который в этом случае должен иметь значительный опыт как в решении задач оптимизации, так и в использовании данного алгоритма оптимизации, должен понимать заложенные в алгоритм идеи, назначения его свободных параметров, знать детали программной реализации алгоритма и т.д.

Современным подходом к настройке алгоритмов оптимизации является подход на основе автоматизированного или автоматического решения задачи настройки как *задачи метаоптимизации* (*М-задачи*). Эта задача заключается в отыскании "оптимальной" стратегии  $B^*$  алгоритма  $a(B)$  при решении данной *базовой задачи оптимизации*  $q(C)$  (*Б-задачи*) или *класса Б-задач оптимизации*  $Q(D_C) = \{q(C) | C \in D_C\}$ , где  $D_C$  — множество допустимых значений компонентов вектора ХП. Настраиваемый алгоритм

оптимизации  $a(B)$  называем *базовым алгоритмом* (*Б-алгоритмом*), а настраивающий — *метаалгоритмом* (*М-алгоритмом*).

Критерий эффективности Б-стратегий называем *индикатором эффективности* и обозначаем  $e = e(q(C), a(B)) = e(C, B)$ . М-задачу редко ставят и решают для одной Б-задачи, напротив, обычно М-задачу решают на некотором классе задач, который мы выше определили как  $Q(D_C)$ . Таким образом, *одноиндикаторную М-задачу* для класса Б-задач  $Q(D_C)$  (*multi-instance problem tuning*) записываем в виде

$$\text{opt}_{B \in D_B} e(Q(D_C), B) = e(D_C, B^*). \quad (2)$$

В зависимости от класса рассматриваемых М-алгоритмов искомой в задаче (2) является *статическая стратегия*  $B^*$  или *динамическая стратегия*  $B^*(t)$ , которая программно или адаптивно изменяется в процессе решения Б-задачи (с ростом номера  $t$  итераций Б-алгоритма).

Подчеркнем, что поскольку, как мы отмечали выше, в состав Б-стратегии могут входить как числовые, так и категориальные параметры, М-задача (2), в отличие от Б-задачи (1), представляет собой, вообще говоря, задачу смешанного программирования. В связи с этим при разработке М-алгоритмов принципиальной является используемая схема кодирования категориальных параметров Б-алгоритма [26].

В качестве индикатора эффективности Б-стратегий на классе задач  $D_C$  обычно используют усредненные значения  $e(D_C, B)$  индикатора  $e(q(C), B)$  на этом классе задач, т. е. полагают, что  $e(D_C, B) = \bar{e}(q(C), B)$ ,  $C \in D_C$ . Иногда рассматривают двумерный вектор индикаторов, первая компонента которого есть  $\bar{e}(D_C, B)$ , а вторая компонента — оценка стандартного отклонения  $\sigma(D_C, B)$  индикатора  $e(q(C), B)$  на том же классе задач. Другими словами, рассматривают М-задачу как мультизадачу М-оптимизации (см. ниже).

**Мультизадачи метаоптимизации.** Выделяем три класса мультизадач М-оптимизации.

**Мультииндикаторные М-задачи.** В этих задачах эффективность Б-стратегий оценивают с помощью вектора индикаторов  $E = (e_1, e_2, \dots, e_{|E|}) = E(D_C, B)$ . Аналогично задаче (2), мультииндикаторную М-задачу рассматриваем в постановке

$$\text{opt}_{B \in D_B} \uparrow E(D_C, B) = E(D_C, B^*), \quad (3)$$

где индекс  $\uparrow$  указывает на многокритериальность постановки задачи;  $B^*$  — набор "оптимальных" стратегий, принадлежащих множеству Парето задачи (3). В качестве компонентов вектора  $E(D_C, B)$  аналогично задаче (2) могут использоваться их средние значения  $\bar{e}_i(D_C, B)$ , стандартные отклонения  $\sigma_i(D_C, B)$ ;  $i \in [1 : |E|]$ .

По общим правилам решения задач многокритериальной оптимизации двумя основными методами решения задачи (3) являются метод скалярной сверки частных индикаторов и метод, предполагающий построение тем или иным способом конечномерной аппроксимации фронта и, тем самым, множества Парето этой задачи. Эти аппроксимации дают возможность ЛПР неформальными либо формализованными методами выбрать "оптимальную" с его точки зрения стратегию, изучить влияние различных компонентов Б-стратегии на эффективность Б-алгоритма, исследовать устойчивость значений индикаторов эффективности к вариациям стратегий и т.д. [27]. Для построения конечномерных аппроксимаций множества и фронта Парето задачи (3) используют как хорошо известные в многокритериальной оптимизации методы (*NSGA-II*, *SPEA-2* и т.д.), так и методы, специально разработанные для решения многоиндикаторной М-задачи.

**Мультиклассовые М-задачи (multi-problem tuning).** М-задачу (2) часто решают не на одном классе задач  $Q(D_C)$ , но на некотором наборе таких классов задач  $Q(D_{C_1}), Q(D_{C_2}), \dots, Q(D_{C_{|D|}})$ , т. е. решают  $|D|$  штук М-задач вида (2):

$$\begin{aligned} \text{opt}_{B \in D_B} e(D_{C_i}, B) &= e(D_{C_i}, B_i^*), \\ D_{C_i} &\in D_C, i \in [1 : |D|]. \end{aligned} \quad (4)$$

Относительно решений М-задачи (4) возникает вопрос: насколько чувствительна каждая из полученных "оптимальных" Б-стратегий  $B_i^* = B^*(D_{C_i})$  к изменению (сужению или расширению) класса задач  $Q(D_{C_i})$ ?

Ответ на этот вопрос отыскивают путем решения многокритериальной мультиклассовой М-задачи

$$\text{opt}_{B \in D_B} \uparrow E(D_C, B) = E(D_C, B^*). \quad (5)$$

Здесь приняты следующие обозначения:  $D_C = \{D_{C_i} \in D_C, i \in [1 : |D|]\}$  — набор, определяющий рассматриваемые классы задач;  $E(D_C, B) =$

$= (e(D_{C_i}, B) = e_i(B), i \in [1 : |D|])$  — векторный индикатор эффективности.

Скалярной свертке индикаторов  $e_i(B)$ ,  $i \in [1 : |D|]$  трудно придать содержательный смысл. Поэтому используют построение конечномерной аппроксимации множества достижимости задачи (5) и/или отыскание таких же аппроксимаций ее множества и фронта Парето.

*Мультибюджетные M-задачи* возникают в связи с тем, что "оптимальная" B-стратегия может сильно зависеть от используемой конфигурации вычислительной системы, т. е., вообще говоря, для разных доступных вычислительных ресурсов нужно отыскивать разные "оптимальные" стратегии. Применительно к каждой данной B-задаче мощность этих ресурсов измеряют максимально допустимым числом испытаний целевой функции  $f(X)$  — *бюджетом* задачи. При небольшом бюджете, обусловленном использованием ЭВМ малой мощности, "оптимальная" B-стратегия обеспечивает преимущественную интенсификацию поиска. Напротив, мощные вычислительные системы позволяют использовать "оптимальные" стратегии, которые реализуют широко диверсифицированный поиск.

Поясним суть подхода к решению мультибюджетной M-задачи на примере решения одной B-задачи  $q(C)$ . Положим, что рассматриваемый B-алгоритм на каждой итерации решения этой задачи требует одинакового числа испытаний целевой функции  $f(X)$ . Тогда, очевидно, бюджет, выраженный в терминах числа испытаний  $f(X)$ , легко пересчитать в бюджет, определенный в терминах числа итераций B-алгоритма  $\hat{t}$ . Обозначаем  $\hat{T} = (\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{|\hat{T}|})$  интересующий исследователя набор бюджетов этого сорта.

Положим, что эффективность B-стратегий оценивается с помощью одного индикатора эффективности  $e(C, B)$ ,  $B \in D_B$ , подлежащего минимизации. Поскольку программное обеспечение глобальной оптимизации строят таким образом, чтобы в процессе итераций лучшие найденные решения B-задачи не терялись, зависимость индикатора  $e(qC, B)$  от текущего номера итерации  $t$  представляет собой невозрастающую *функцию сходимости итерационного процесса*, т. е. для каждой из стратегий  $B \in D_B$  имеет вид, представленный на рис. 1. Заметим, что фиксация этой зависимости требует пренебрежимо малых вычислительных затрат.

Рассмотрим некоторые допустимые B-стратегии  $B_1, B_2, B_3$ . Рис. 1 показывает возможный

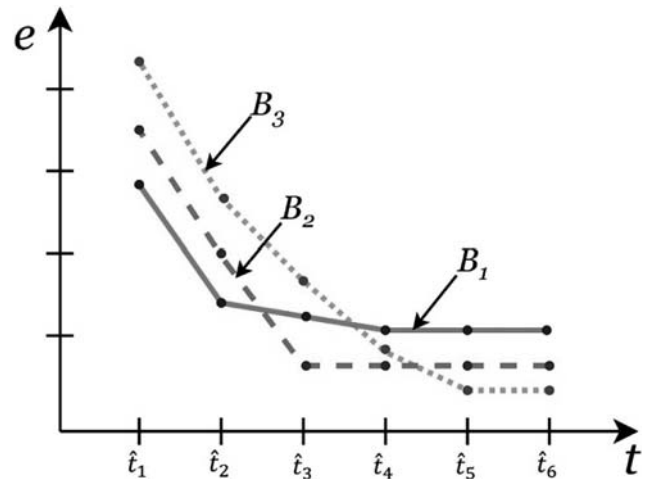


Рис. 1. Зависимость лучшего найденного значения индикатора эффективности  $e = e(C, B)$  от текущего номера  $t$  итераций B-алгоритма для некоторых допустимых стратегий  $B_1, B_2, B_3$ :  $\hat{t}_1, \dots, \hat{t}_6$  — бюджеты

характер сходимости итерационных процессов, порожденных рассматриваемым B-алгоритмом, для этих стратегий. Из рис. 1 следует, что "оптимальная" B-стратегия  $B^*$  для разных бюджетов имеет вид

$$B^* = \begin{cases} B_1, \hat{t} = \hat{t}_1, \hat{t}_2; \\ B_2, \hat{t} = \hat{t}_3, \hat{t}_4; \\ B_3, \hat{t} = \hat{t}_5, \hat{t}_6. \end{cases}$$

По общим правилам при решении мультибюджетной M-задачи на классе задач  $Q(D_C)$  в качестве индикатора эффективности B-стратегий можно использовать, например, среднее значение индикатора  $e(C, B)$  на этом классе задач.

Заметим, что постановка M-задачи как мультибюджетной позволяет ввести новые метрики качества B-стратегий по следующей схеме. Для каждого бюджета  $\hat{t}_i, i \in [1 : |\hat{T}|]$ , ранжируем рассматриваемый набор B-стратегий  $\{B_j \in D_B\}$  по возрастанию индикатора  $e(C, B)$ , так что ранг стратегии  $B_k$  равен  $r(\hat{t}_i, B_k) = 1$ , если для бюджета  $\hat{t}_i$  индикатор  $e(C, B_k)$  имеет минимальное значение; аналогично  $r(\hat{t}_i, B_l) = 2$ , где  $B_l \in \{B_j\} \setminus B_k$  и т.д. Каждую из стратегий набора  $\{B_j\}$  характеризуем числом бюджетов  $\hat{t}_i, i \in [1 : |\hat{T}|]$ , при которых она является лучшей, т. е. имеет ранг  $r = 1$ . Мерой качества стратегии  $B_j$  может служить также площадь под непрерывным графиком  $e(C, B_j, \hat{t})$  (меньшая площадь означает более быструю сходимость соответствующего итерационного процесса и более высокое качество стратегии).

Как и в одноиндикаторных М-задачах, искомой в рассмотренных мультизадачах М-оптимизации могут являться статическая  $B^*$  или динамическая  $B^*(t)$  стратегии.

Введем в заключение этого раздела следующие обозначения:  $1M^2$  — одна из рассмотренных мультизадач М-оптимизации;  $2M^2$  — комбинированная мультизадача М-оптимизации, объединяющая любые две из этих мультизадач;  $3M^2$  — аналогичная задача, включающая в себя все три мультизадачи.

## 2. Методы ландшафтного анализа целевой функции

Для определения значений апостериорных ХП целевой функции методами ЛА необходима обучающая выборка — набор точек  $X \in D_X$  и соответствующих значений целевой функции  $f(X)$ . Найденные значения ХП определяются не только ландшафтом функции  $f(X)$ , но и методом планирования эксперимента, использованным при формировании выборки [6]. Поэтому предпочтение отдают адаптивным методам планирования эксперимента [28], которые по сравнению с неадаптивными методами позволяют получить более точные значения ХП целевой функции за меньшее число испытаний.

Классический метод ландшафтного анализа [25] предлагает в общей сложности 50 числовых признаков ландшафта функции  $f(X)$ , сгруппированных в шесть так называемых свойств: выпуклость, степень кривизны, у-распределение, ярусность, мультимодальность, метамодельные свойства. Здесь под у-распределением понимают статистические оценки плотности вероятности распределения значений  $f(X)$  в точках обучающей выборки, число экстремумов функции плотности вероятности, степень близости этой функции к нормальному закону распределения. Ярусность функции  $f(X)$  оценивают как вероятность того, что ее значения превышают заданное значение. Метамодельные признаки  $f(X)$  вычисляют на основании результатов оценки точности ее линейных или квадратичных регрессионных М-моделей, построенных на используемой обучающей выборке. Подчеркнем, что определение некоторых из указанных свойств, например свойства мультимодальности,

требует проведения дополнительных испытаний функции  $f(X)$  в новых точках области  $D_X$ , что накладывает сильные ограничения на применение данного метода ЛА для исследования задач, имеющих высокую вычислительную сложность целевой функции.

Метод клеточного отображения (Cell Mapping) [29]. В этом случае область поиска  $D_X$  по каждому из измерений пространства  $\mathbb{R}^{|X|}$  равномерно разбивают на подобласти, называемые клетками. Для каждой клетки вычисляют ХП функции  $f(X)$ , оценивающие выпуклость, однородность градиента функции и угол между векторами, которые соединяют центр клетки с точками выборки внутри нее, имеющими наибольшее и наименьшее значение  $f(X)$  (рис. 2). В случае, если функция  $f(X)$  имеет мало локальных оптимумов, суммарный разброс вычисленных значений признаков по всем клеткам является небольшим, поскольку соответствующие точки находятся на границах этих клеток.

Метод обобщенного клеточного отображения (Generalized Cell Mapping) [29]. Область  $D_X$  аналогичным образом разбивают на клетки. Каждой клетке ставят в соответствие одно значение функции  $f(X)$ , например, в точке  $X$ , ближайшей к центру клетки. ХП ландшафта функции  $f(X)$  вычисляют на основе оценки вероятностей перехода между клетками с помощью поглощающих цепей Маркова.

Метод барьерных деревьев. После дискретизации области поиска  $D_X$  методом обобщенного клеточного отображения применяют барьерные деревья (Barrier Trees) [30]. Идея заключается в представлении полученного множества клеток

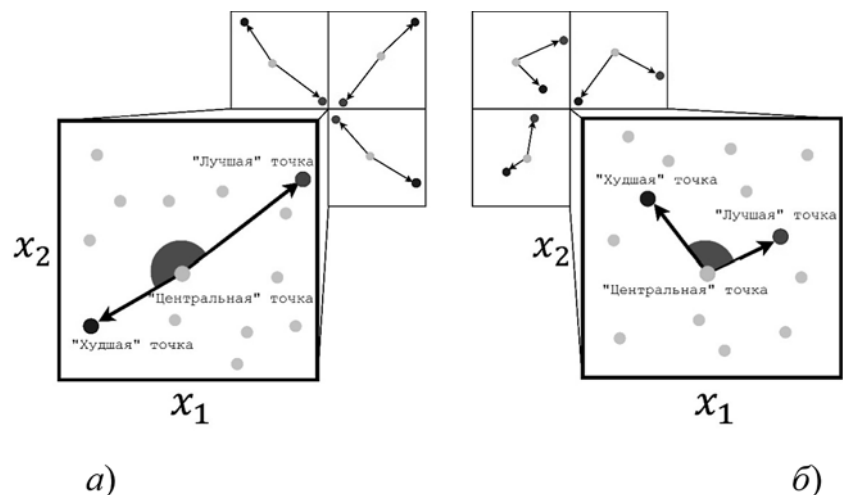


Рис. 2. Метод клеточного отображения: характерные точки клетки для унимодальной (а) и мультимодальной (б) функций;  $|X| = 2$

в виде дерева, в котором клетки, содержащие локально "оптимальные" точки целевой функции  $f(X)$ , соответствуют листьям. Значения ХП функции  $f(X)$  определяют на основе оценок различных параметров построенного дерева: число листьев; высота дерева; статистические оценки разности значений  $f(X)$  в клетках, соответствующих соседним узлам дерева, и т.д.

Метод информационного содержания (Information Content) [31] позволяет оценить численные характеристики зашумленности и мульти-модальности ландшафта целевой функции  $f(X)$ . Сначала точки  $X_i, i \in [1 : l]$ , обучающей выборки упорядочивают тем или иным образом и вычисляют перепад значений  $f(X)$  между соседними точками выборки. Затем последовательность значений  $f(X_i) = f_i$  преобразуют в символичный набор  $S(\varepsilon) = \{s_1, \dots, s_{l-1}\}$  по правилу

$$s_i = \begin{cases} \bar{1}, & \delta_i < -\varepsilon; \\ 0, & |\delta_i| \leq \varepsilon; \\ 1, & \delta_i > \varepsilon, \end{cases} \quad \delta_i = \frac{f_{i+1} - f_i}{\|X_{i+1} - X_i\|}, \quad i \in [1 : l - 1],$$

где  $\varepsilon$  — параметр, определяющий чувствительность метода;  $\|\cdot\|$  — евклидова норма. Полученную последовательность символов используют для вычисления искомого ХП на основании значений функции информационного содержания

$$ic(\varepsilon) = - \sum_{a \neq b} p_{ab}(\varepsilon) \log_6 p_{ab}(\varepsilon), \quad a, b \in \{\bar{1}, 0, 1\},$$

где  $ab$  — блок последовательных символов из набора  $S(\varepsilon)$ , например  $\bar{1}1, 01, 11$ ;  $p_{ab}$  — вероятность обнаружения блока  $ab$  в наборе  $S(\varepsilon)$ .

Известно некоторое число других методов ЛА. Например, метод *Nearest-Better Clustering* [32, 33] позволяет идентифицировать единичные большие скачки мультимодальной целевой функции  $f(X)$ . Метод главных компонент [34] определяет ландшафт числом главных компонент рассматриваемой выборки, при котором значение остаточной (объяснительной) дисперсии не превышает заданного значения. Предложены также ХП на основе оценки дисперсии попарных расстояний между всеми точками выборки и лучшими из этих точек [14].

### 3. Метаоптимизация

#### 3.1. Общие сведения

В соответствии с классификацией, предложенной в работе [35], на верхнем уровне иерархии выделяют две группы методов М-оптимизации — методы *настройки* параметров и мето-

ды *управления* параметрами (рис. 3, см. вторую сторону обложки). При настройке параметров выбранную стратегию не меняют в ходе решения Б-задачи. Информацию об эффективности Б-алгоритма с данной стратегией получают после окончания решения этой задачи. В случае управления параметрами стратегию варьируют в процессе решения Б-задачи. Детальное сравнение этих групп методов М-оптимизации выполнено в работе [36].

Методы настройки параметров являются более универсальными в сравнении с методами управления параметрами, поскольку не требуют доработки Б-алгоритма, который в этом случае рассматривается как "черный ящик". Все пользователи Б-алгоритма после окончания процесса настройки получают "оптимальную" стратегию для решения определенного класса Б-задач, что избавляет их от дальнейших вычислительных затрат на М-оптимизацию. Методы настройки позволяют дополнять накопленный опыт решения Б-задач, уточняя найденные "оптимальные" Б-стратегии.

Выделяют два основных подхода к настройке параметров [37]. Целью первого подхода является быстрое отыскание "оптимальной" Б-стратегии путем *интенсификации* поиска в пространстве стратегий. Цель второго подхода состоит в получении информации о поведении Б-алгоритма в процессе решения им Б-задач для более глубокого понимания особенностей функционирования этого алгоритма. Достижение последней цели требует обширного исследования пространства Б-стратегий, т. е. *диверсификации* поиска.

В работе [37] выделены следующие основные подходы к "одноиндикаторной" настройке параметров (рис. 4, см. вторую сторону обложки), отличающиеся, прежде всего, соотношением интенсификационной и диверсификационной составляющих поиска "оптимальной" Б-стратегии:

- *стохастические поисковые методы*, "заточенные" под быстрое отыскание "оптимальных" стратегий;
- *методы сэмплирования (sampling)*, позволяющие провести обширное исследование пространства стратегий;
- *скрининговые методы*, содержащие различные техники раннего выявления перспективных стратегий.
- *метамодельные методы*, использующие суррогатную модель индикатора эффективности стратегий (*М-модель*) и ориентированные на интенсификацию поиска "оптимальных" стратегий;



- *комбинированные методы.*

Выделяем также следующие классы методов настройки параметров:

- методы, ориентированные на решение мультизадач М-оптимизации;
- методы настройки параметров для параллельных вычислительных систем.

В терминах [37] различают *итеративные* и *не итеративные* методы настройки. В случае не итеративной настройки тем или иным образом генерируют набор исследуемых стратегий, из числа которых и выбирают "оптимальную" для решения рассматриваемого класса Б-задач. При итеративной настройке параметров (штриховые стрелки на рис. 4) набор Б-стратегий, из числа которых выбирают "оптимальную", не фиксирован. Новые стратегии генерируют на основе информации, поступающей в процессе настройки.

Методы однократной настройки могут быть как итеративными, так и не итеративными, в то время как методы перманентной настройки являются строго итеративными. Наоборот, итеративные методы можно использовать как для однократной, так и для перманентной настройки Б-алгоритмов. Не итеративные методы ориентированы на однократную настройку.

В пп. 3.2—3.6 последовательно рассматриваем представленные выше "одноиндикаторные" методы настройки, а в пп. 3.7, 3.8 — методы настройки для решения мультизадач М-оптимизации и параллельные методы настройки.

### 3.2. Стохастические поисковые методы

Исторически первым представителем стохастических методов М-оптимизации является метаэволюционный алгоритм (*meta-EA*), предложенный в 1978 г. [38]. Параметры в этом методе настраиваются с помощью эволюционного алгоритма. Широкое исследование эффективности метода не было выполнено в силу его относительно высокой вычислительной сложности. Обширные исследования были выполнены с использованием близкого метагенетического алгоритма (*meta-GA*). Здесь стратегиям Б-алгоритма поставлены в соответствие особи М-алгоритма, а индикатору эффективности стратегий — фитнес-функции М-алгоритма. Заметим, что такая интерпретация М-задачи позволяет использовать для настройки параметров любой эволюционный алгоритм. Исследования показали высокую эффективность данного М-алгоритма настройки параметров [39].

Метод *FocusedILS* [40] использует в качестве М-алгоритма гибридизацию алгоритмов *эволюционной стратегии* и локальной оптимизации. Авторы метода предложили специальную технику для сравнения эффективности испытываемых и выявления "оптимальных" Б-стратегий, позволяющую уменьшить общее число испытаний стратегий (обращений к Б-алгоритму). Похожее решение предложено в работе [41], авторы которой добавили скрининговый метод *гонки* (см. п. 3.4). Недостатком подобных гибридизаций является усложнение М-алгоритма и, как следствие, рост вычислительных затрат на М-оптимизацию.

Метод *SMAC* (*Sequential Model-based Algorithm Configuration*) [42] представляет собой модифицированный метод *FocusedILS*. Основной особенностью *SMAC* является использование М-модели фитнес-функции М-задачи для локального уточняющего поиска "оптимальной" Б-стратегии. Перспективность стратегии определяют на основе модели случайного леса (*Random Forest Model*) [43].

### 3.3. Методы сэмплирования

Подход к М-оптимизации на основе сэмплирования ориентирован на сокращение общего числа испытаний стратегий (в ущерб, естественно, широте поиска). Методы данного класса чаще используют на стадии инициализации поиска "оптимальной" стратегии с помощью, например, метамодельных методов, нежели в качестве самостоятельных М-методов. Наиболее известны следующие не итеративные методы, реализующие данный подход: метод на основе *латинского квадрата* (*Latin-Square*) [44]; метод на основе *алгоритма Тагучи* (*Taguchi Orthogonal Arrays*) [45]. Известными примерами итеративных методов сэмплирования являются методы *CALIBRA* [46] и *Empirical Modeling of Genetic Algorithms* [44].

В работе [47] предложены следующие простые итеративные методы настройки, использующие сэмплирование: метод на основе паттернов (*Pattern Search, PS*); метод локальной унимодальной выборки (*Local Unimodal Sampling, LUS*). Результаты проведенного авторами исследования не показали заметного снижения эффективности "оптимальных" Б-стратегий, найденных этими методами, по сравнению со стратегиями, полученными с помощью некоторых других значительно более ресурсоемких М-алгоритмов.

### 3.4. Скрининговые методы

Целью скрининговых методов М-оптимизации является выявление "оптимальной" стратегии за минимальное число запусков Б-алгоритма. Скрининговые методы сформировались под влиянием методов ранжирования и выбора лучшей из имеющихся имитационных моделей исследуемой системы по степени достоверности и точности результатов моделирования [48]. Б-стратегии в этом случае ставятся в соответствие имитационная модель, а решению Б-задачи — испытание этой модели. М-алгоритм соответствует методу отбора имитационных моделей.

В работе [49] выполнено сравнение эффективности наиболее известных скрининговых методов: *Interactive Analysis* [50]; *Ranking and Selection* [51]; *Multiple Comparison Procedures* [52]; *Fully Sequential Indifference-zone Selection procedure* [53]. Показано, что указанные методы значительно отличаются необходимым числом испытаний исследуемых моделей и гарантиями отбора лучшей из них.

Скрининговые методы основаны на допущении, что результаты стохастической имитации распределены по нормальному закону. Эти допущения являются слабо обоснованными, но могут быть в значительной мере удовлетворены путем перегруппирования результатов стохастических испытаний [49]. Преимуществом скрининговых методов можно считать определенные гарантии локализации "оптимальной" Б-стратегии. Уровень доверия к "оптимальности" полученной стратегии можно оценить, например, с помощью метода *Multiple Comparison Procedures* [48, 49].

Известным представителем итеративных скрининговых методов является метод *I/F-RACE* [54], построенный на основе метода *F-RACE* [60] и сочетающий в себе техники скрининга и локального уточняющего поиска. Область допустимых стратегий  $D_B$  в этом методе покрывают сеткой с относительно небольшим числом узлов. С помощью гонок [55] отбирают наиболее перспективные Б-стратегии, на основании которых строят аппроксимацию функции плотности вероятности локализации "оптимальных" стратегий в подобластях множества  $D_B$ . Полученную вероятностную М-модель используют для генерации новых Б-стратегий.

Применительно к настройке параметров часто используют подход на основе гонок, который впервые был предложен в методе *Hoeffding Races* [55]. В сущности, этот подход незначи-

тельно отличается от рассмотренных выше методов, но позволяет исследовать большее число Б-стратегий. В основе подхода лежит правило изменения числа запусков Б-алгоритма с каждой из стратегий (числа испытаний стратегии) для оценки ее эффективности. Начинают гонки с малого числа испытаний, по результатам которых отбрасывают явно неэффективные стратегии. В последующих итерациях гонок постепенно увеличивают число испытываемых стратегий для более точной оценки их эффективности. В результате более перспективные стратегии исследуются чаще, чем малоэффективные. Преимущество гонок состоит в том, что они не требуют каких-либо допущений о распределении значений используемого индикатора эффективности Б-алгоритма (см. методы *Hoeffding Races* [55], *F-RACE* [56]).

### 3.5. Метамоделные методы настройки

Данный класс М-методов настройки основан на использовании аппроксимирующей (суррогатной) модели (М-модели) индикатора эффективности  $e(C, B)$ , построение которой имеет целью уменьшить число вычислительно емких испытаний Б-стратегий. Аналогично методу ЛА при использовании этого класса методов настройки качество предсказаний М-модели напрямую зависит от метода планирования эксперимента, с помощью которого осуществляется генерация соответствующей обучающей выборки [57].

Часто для построения М-модели используют метод регрессии [58, 59]. Точность модели, а значит, и качество найденных "оптимальных" Б-стратегий оказываются в этом случае относительно низкими. Итеративные метамоделные методы позволяют и при использовании регрессионной М-модели найти приемлемые "оптимальные" Б-стратегии.

Одним из наиболее известных метамоделных методов настройки является процедура Коя (*Coy's procedure*) [60]. Основная идея процедуры состоит в использовании методов локального поиска для уточнения Б-стратегий, полученных с помощью М-модели. На первом этапе область  $D_B$  дискретизируют и в узлах полученной сетки вычисляют значения индикатора эффективности  $e(C, B)$ . На основании полученных данных строят линейную регрессионную М-модель этого индикатора. На втором этапе с помощью построенной модели генерируют и испытывают Б-стратегии с использованием метода наискорейшего спуска.

*Метод SPO (Sequential Parameter Optimization)*, в отличие от двухступенчатой процедуры Коя, постоянно обновляет М-модель [61, 62]. Каждая итерация М-алгоритма в этом случае включает в себя следующие действия: обновление М-модели на основе результатов предыдущей итерации; генерация набора Б-стратегий и прогнозирование их эффективности с использованием текущей М-модели; испытания наиболее перспективных стратегий.

Качество "оптимальных" Б-стратегий, найденных метамодельными методами настройки, во многом зависит от типа используемой М-модели. Исследование эффективности различных типов этих моделей (суррогатных функций), способов их построения и использования в целях настройки параметров Б-алгоритма выполнено в работе [63].

### 3.6. Комбинированные методы

*Метод SEPaT (Simple Evolutionary Parameter Tuning)* [26] первоначально позиционировался авторами как стохастический М-метод. Однако используемый методом *SEPaT* подход к оценке эффективности Б-стратегий может применяться в любых других одноиндикаторных М-методах. Метод *SEPaT* реализует различные правила быстрого отбора эффективных Б-стратегий, в том числе стратегий, содержащих категориальные параметры. Эффективность стратегий вычисляют на основе среднего и стандартного отклонений индикатора эффективности  $e(D_C, B)$ . Предполагается, что одна Б-стратегия лучше другой, если она эффективнее в среднем и имеет меньшее стандартное отклонение значений индикатора эффективности. В противном случае, лучшую стратегию выбирают на основе результатов теста Велча (*Welch's t-test* [64]), т. е. на основе оценки величины

$$e'(q_i, B_1, B_2) = \sum_{i=1}^n \frac{\bar{e}(q_i, B_1) - \bar{e}(q_i, B_2)}{\sqrt{\frac{\sigma^2(q_i, B_1) - \sigma^2(q_i, B_2)}{m}}},$$

$$q_i = q_i(C) \in Q(D_C),$$

где  $B_1, B_2$  — сравниваемые допустимые стратегии;  $\bar{e}(q_i, B)$ ,  $\sigma(q_i, B)$  — среднее значение и стандартное отклонение индикатора эффективности  $e(q, B)$ , полученные на основе решения  $m$  раз Б-задачи  $q_i$ . В случае минимизации индикатора эффективности  $e(q, B)$  стратегия  $B_1$  лучше стратегии  $B_2$ , если  $e'(q, B_1, B_2) < 0$ . Если необходим анализ эффективности более

двух стратегий, то их сравнивают попарно каждую с каждой, и стратегию с наименьшим числом поражений признают лучшей, а в случае равенства числа проигрышей сравнивают число выигрышей. Таким образом, найденная "оптимальная" стратегия оказывается наиболее эффективной в среднем для всех рассматриваемых Б-задач класса  $D_C$ . Более детально метод рассмотрен в работе [65], где его сравнивают с методами гонок [55] и *ParamILS* [66].

### 3.7. Методы решения мультизадач М-оптимизации

*Метод M-FETA (Multi-Function Evolutionary Tuning Algorithm)* [27] использует специализированный алгоритм мультииндикаторной настройки параметров. Основным преимуществом алгоритма является наличие специальных операторов, понижающих число требуемых запусков Б-алгоритма для исследования каждой из испытываемых Б-стратегий. С этой целью эффективность некоторых стратегий оценивают с помощью значений индикаторов эффективности "соседних" стратегий; в перспективных подобластях множества  $D_B$  генерируют большее число стратегий, что уменьшает расстояние между "соседними" перспективными стратегиями и позволяет уточнить оценки их эффективности.

*Метод REVAC (Relevance Estimation and Value Calibration of Parameters)* в исходном варианте относится к эвристическим итеративным стохастическим поисковым методам настройки [67], однако известна его мультииндикаторная модификация. В этой модификации фронт Парето Б-стратегий не строят, но используют оригинальную методику многоиндикаторного ранжирования стратегий на основании оценки их эффективности при решении различных Б-задач. Метод позволяет использовать гонки [55] в целях уменьшения числа испытаний Б-стратегий.

*Метод Bonesa* [68] в своей основе является итеративным методом настройки параметров, использующим М-модель. Для уменьшения числа запусков стохастических Б-алгоритмов с каждой из испытываемых Б-стратегий используется фильтр Гаусса [69], который уменьшает уровень зашумленности индикатора эффективности. Сравнение эффективности стратегий выполняют с помощью гонок [55]. Для построения фронта Парето метод *Bonesa* использует алгоритм *SPEA2* [70].

*Метод EMOPaT (Evolutionary Multi-Objective)* [26] является мультиклассовой модификацией

метода *SEPaT*. В качестве М-алгоритма использован алгоритм *NSGA-II* [71]. Метод включает в себя анализ Парето-оптимальных решений с помощью простого и эффективного подхода *innovation (innovation through optimization)* [72]. Компонентами вектора Б-стратегий в методе *EMOPaT* могут быть категориальные параметры. Схема кодирования этих параметров представлена в работе [26].

*Метод Flexible Budget* [73] относится к классу мультибюджетных методов М-оптимизации. Результатом настройки является набор Б-стратегий, эффективных при различных бюджетах.

### **3.8. Методы для параллельных вычислительных систем**

*Метод ReACT (Real-time Algorithm Configuration through Tournaments)* [74] позволяет одновременно испытывать несколько Б-стратегий посредством гонок с использованием многопроцессорных систем. На каждом из доступных процессоров запускают экземпляр Б-алгоритма с одной из допустимых стратегий. После завершения всех запущенных Б-алгоритмов определяют стратегию-победителя. Если по истечении заданного ограничения по времени ни один из Б-алгоритмов не завершится, то победителем считают стратегию с лучшим достигнутым значением целевой функции  $f(X)$ . Как и *SEPaT*, данный метод может быть скомбинирован с любыми методами одноиндикаторной настройки.

## **4. Современное программное обеспечение глобальной оптимизации**

### **4.1. Базовая оптимизация**

Значительное число методов глобальной оптимизации, не использующих аналитические или численные производные в процессе поиска, имеют открытые реализации: *Bound Optimization BY Quadratic Approximation (BOBYQA)* [75]; *Design Analysis Kit for Optimization and Terascale Applications (DAKOTA)* [76]; *GLOBAL* [77]; *Hybrid Optimization Parallel Search PACKage (HOPSPACK)* [78]; *NEWUOA* [79]; *NOMAD* [80]; *SID-PSM* [81] и т.д.

Для решения Б-задач, имеющих высокую вычислительную сложность, разработаны специальные программные комплексы, представленные ниже.

*Платформа modeFRONTIER*, помимо традиционных инструментов для оптимизации,

включает в себя большой набор гибридных алгоритмов [82], единственным свободным параметром которых является максимальное число испытаний целевой функции. Алгоритм *FAST* использует *RSM* модели для исследования наиболее перспективных областей пространства поиска. Алгоритм *HYBRID* сочетает устойчивость генетических алгоритмов и точность градиентных. В рамках алгоритма *SAnGeA* реализован подход, который позволяет определить наиболее значимые компоненты вектора варьируемых параметров  $X$ . Алгоритм *piOPT* комбинирует методы локального, глобального поиска и метамоделирования.

*Программный продукт HEEDS Professional* реализует проприетарный гибридный адаптивный алгоритм *SHERPA* [83]. Алгоритм использует несколько методов локальной и глобальной оптимизации, автоматически модифицируя их параметры по мере поступления информации о ландшафте целевой функции  $f(X)$ . Единственным свободным параметром алгоритма является максимально допустимое число испытаний  $f(X)$ .

*Модуль Generic Tool for Optimization (GTOpt)* [84], поставляемый компанией *Datadvice*, спроектирован для решения сложных прикладных инженерных оптимизационных задач. Модуль содержит оригинальные высокоэффективные алгоритмы одно- и многокритериальной оптимизации, использующие метамоделей целевых функций

*Программный комплекс IOSO NM* реализует алгоритм *Indirect Optimization on the basis of Self-Organization (IOSO)* [85], разработанный компанией *Сигма Технологии*. Программный комплекс пригоден для решения широкого спектра оптимизационных задач, в том числе с одновременным использованием нескольких моделей оптимизируемого объекта. Алгоритм *IOSO* использует эволюционные алгоритмы с самоорганизацией. Высокая эффективность алгоритма обеспечивается набором встроенных правил адаптивного выбора алгоритма оптимизации.

### **4.2. Программное обеспечение для настройки параметров**

*Универсальный программный продукт SMAC* реализует одноименный метод настройки параметров Б-алгоритмов, а также иных параметризованных объектов, например имитационных моделей, алгоритмов машинного обучения

и т.д. [86]. Продукт *SMAC* не имеет графического интерфейса, запуск М-алгоритма приходится проводить с помощью командной строки. Пользователь должен предоставить программе бинарный файл Б-алгоритма, конфигурационные файлы с описанием стратегий и индикатора эффективности Б-алгоритма. Результаты настройки также выдаются в виде файла.

Подобный способ взаимодействия с пользователем реализован и в программном комплексе *ParamILS* [87]. Основные различия между *SMAC* и *ParamILS* заключаются в следующем.

- Комплекс *ParamILS* позволяет использовать только категориальные свободные параметры Б-алгоритма, в то время как *SMAC* поддерживает и числовые параметры. Также *SMAC* предоставляет более широкий выбор индикаторов эффективности Б-стратегий.
- Программа *SMAC* имеет возможность гибкой настройки временных и вычислительных ограничений М-алгоритма и позволяет использовать дополнительную информацию о ХП исследуемых Б-задач.

Программный комплекс *DGGA Algorithm Configurator* реализует метод настройки *GGA* [88], поддерживающий как категориальные, так и числовые параметры Б-алгоритма. Метод *GGA* использует технику гонок для поиска "оптимальных" Б-стратегий. Комплекс ориентирован на решение М-задач с применением параллельных вычислительных систем. Аналогично программным комплексам *SMAC*, *ParamILS*, для конфигурирования М-алгоритма и описания исходных данных М-задачи используются файлы специальной структуры.

Программный пакет *irace* [89] разработан на основе расширенного итеративного метода настройки *I/F-Race* [54], использующего гонки. Пакет предоставляет возможность параллельного исполнения М-алгоритма, задания начального набора стратегий, указания признаков "плохих" стратегий, которые должны быть исключены из процедуры поиска. Пакет широко применяют для настройки Б-алгоритмов непрерывной и дискретной оптимизации [90], в том числе алгоритмов многокритериальной оптимизации [91].

## Заключение

Современное состояние техник глобальной оптимизации позволяет сделать следующие выводы.

В целях сокращения вычислительных затрат в программном обеспечении "промышленной" глобальной оптимизации возрастает поддержка предварительного анализа постановки задачи на основе использования известных и новых алгоритмов анализа данных, понижения размерности пространства поиска, ландшафтного анализа целевой функции.

Имеет место тенденция преимущественного использования гибридных адаптивных алгоритмов оптимизации, объединяющих несколько стохастических алгоритмов глобальной оптимизации и алгоритмов локальной оптимизации (мультимемевые алгоритмы).

Повышается "интеллектуальность" программных комплексов глобальной оптимизации, в том числе за счет реализации в них методов метаоптимизации базовых алгоритмов оптимизации. Если в настоящее время, как правило, используются методы однократной настройки параметров этих алгоритмов, то перспективным является использование методов перманентной настройки, а также адаптивных и самоадаптивных методов управления параметрами. Вместе с тем сейчас, почти без исключений, мультизадачи метаоптимизации решают только по отдельности, т. е. как  $1M^2$ -задачи. Актуальной является проблема совместного решения таких задач (т. е. решения  $2M^2$ - и  $3M^2$ -задач). При этом возникают сложные задачи многокритериального принятия решений, требующие одновременного анализа более одного многомерного фронта (множества) Парето. В целом современные методы многокритериального принятия решений еще недостаточно широко используются в процессе решения задач метаоптимизации. Например, перспективным, на наш взгляд, является использование подходов, основанных на выявлении так называемой функции предпочтений ЛПП [92, 93].

Для решения современных практически важных задач оптимизации все более широко используют параллельные вычислительные системы, имеющие разные архитектуры — вычислительные кластеры, системы с общей памятью, системы на основе графических процессорных устройств, слабосвязанные вычислительные системы типа GRID-систем и т.д. В связи с этим актуальными являются проблемы разработки методов, алгоритмов и соответствующего программного обеспечения для решения базовых задач оптимизации и задач метаоптимизации, ориентированных на указанные классы параллельных систем.

## Список литературы

1. **Shan S., Wang G. G.** Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions // *Structural and Multidisciplinary Optimization*. 2010. Vol. 41, N. 2. P. 219–241.
2. **Wolpert D. H., Macready W. G.** No free lunch theorems for optimization // *IEEE transactions on evolutionary computation*. 1997. Vol. 1, N. 1. P. 67–82.
3. **Rios L. M., Sahinidis N. V.** Derivative-free optimization: a review of algorithms and comparison of software implementations // *Journal of Global Optimization*. 2013. Vol. 56, N. 3. P. 1247–1293.
4. **Kirkpatrick S.** Optimization by simulated annealing: Quantitative studies // *Journal of statistical physics*. 1984. Vol. 34, N. 5–6. P. 975–986.
5. **Liepins G. E., Hilliard M. R.** Genetic algorithms: Foundations and applications // *Annals of operations research*. 1989. Vol. 21, N. 1. P. 31–58.
6. **Hansen N.** The CMA evolution strategy: a comparing review // *Towards a new evolutionary computation*. Springer, Berlin Heidelberg, 2006. P. 75–102.
7. **Poli R., Kennedy J., Blackwell T.** Particle swarm optimization // *Swarm intelligence*. 2007. Vol. 1, N. 1. P. 33–57.
8. **Karaboga D., Basturk B.** A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm // *Journal of global optimization*. 2007. Vol. 39, N. 3. P. 459–471.
9. **Dorigo M., Birattari M., Stutzle T.** Ant colony optimization // *IEEE computational intelligence magazine*. 2006. Vol. 1, N. 4. P. 28–39.
10. **Jones D. R., Schonlau M., Welch W. J.** Efficient global optimization of expensive black-box functions // *Journal of Global optimization*. 1998. Vol. 13, N. 4. P. 455–492.
11. **Cox D. D., John S.** SDO: A Statistical Method for Global Optimization // *Multidisciplinary Design Optimization: State of the Art*. 1997. Vol. 80. P. 315–329.
12. **Huyer W., Neumaier A.** SNOBFIT — Stable noisy optimization by branch and fit // *ACM Transactions on Mathematical Software (TOMS)*. 2008. Vol. 35, N. 2. P. 1–21.
13. **Hansen N.** et al. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Diss. INRIA, 2010.
14. **Decison Tree for Optimization Software.** URL: <http://plato.asu.edu/guide.html> (дата обращения: 25.01.2018).
15. **Momin J., Yang X. S.** A literature survey of benchmark functions for global optimization problems // *Journal of Mathematical Modelling and Numerical Optimisation*. 2013. Vol. 4, N. 2. P. 150–194.
16. **Gould N. I. M., Orban D., Toint P. L.** CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization // *Computational Optimization and Applications*. 2015. Vol. 60, N. 3. P. 545–557.
17. **Global Optimization Test Problems.** URL: <http://www.mat.univie.ac.at/~neum/glopt/test.html> (дата обращения: 25.01.2018).
18. **Dolan E. D., Moré J. J., Munson T. S.** Benchmarking optimization software with COPS 3.0 // *Argonne National Lab, IL (US)*. 2004. No. ANL/MCS-TM-273.
19. **Floudas C. A.** et al. Handbook of test problems in local and global optimization // *Springer Science & Business Media*. 2013. Vol. 33.
20. **Beiranvand V., Hare W., Lucet Y.** Best practices for comparing optimization algorithms // *Optimization and Engineering*. 2017. Vol. 18, N. 4. P. 815–848.
21. **Dolan E. D., Moré J. J.** Benchmarking optimization software with performance profiles // *Mathematical programming*. Vol. 91, N. 2. P. 201–213.
22. **Gould N., Scott J.** A Note on Performance Profiles for Benchmarking Software // *ACM Transactions on Mathematical Software (TOMS)*. 2016. Vol. 43, N. 2. P. 15.
23. **Moré J. J., Wild S. M.** Benchmarking derivative-free optimization algorithms // *SIAM Journal on Optimization*. 2009. Vol. 20, N. 1. P. 172–191.
24. **Bongartz I.** et al. CUTE: Constrained and unconstrained testing environment // *ACM Transactions on Mathematical Software (TOMS)*. 1995. Vol. 21, N. 1. P. 123–160.
25. **Mersmann O.** et al. Exploratory landscape analysis // *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011. P. 829–836.
26. **Ugolotti R.** Meta-optimization of Bio-inspired Techniques for Object Recognition. Diss. Università di Parma. Dipartimento di Ingegneria dell'Informazione, 2015.
27. **Smit S. K., Eiben A. E., Szilávik Z.** An MOEA-based Method to Tune EA Parameters on Multiple Objective Functions // *IJCCI (ICEC)*. 2010. P. 261–268.
28. **Li G., Aute V., Azarm S.** An accumulative error based adaptive design of experiments for offline metamodeling // *Structural and Multidisciplinary Optimization*. 2010. Vol. 40, N. 1. P. 137–155.
29. **Kerschke P.** et al. Cell mapping techniques for exploratory landscape analysis // *EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*. Springer, Cham. 2014. P. 115–131.
30. **Flamm C.** et al. Barrier trees of degenerate landscapes // *Zeitschrift für physikalische chemie*. 2002. Vol. 216, N. 2. P. 155.
31. **Muñoz M. A., Kirley M., Halgamuge S. K.** Exploratory landscape analysis of continuous space optimization problems using information content // *IEEE Transactions on Evolutionary Computation*. 2015. Vol. 19, N. 1. Pp. 74–87.
32. **Preuss M.** Improved Topological Nicheing for Real-Valued Global Optimization // *European Conference on the Applications of Evolutionary Computation*. Springer, Berlin, Heidelberg. 2012. P. 386–395.
33. **Kerschke P.** et al. Detecting funnel structures by means of exploratory landscape analysis // *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM. 2015. P. 265–272.
34. **Munoz M. A., Smith-Miles K.** Effects of function translation and dimensionality reduction on landscape analysis // *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE. 2015. P. 1336–1342.
35. **Eiben A. E., Michalewicz Z., Schoenauer M., Smith J. E.** Parameter Control in Evolutionary Algorithms // *Parameter setting in evolutionary algorithms*. Springer Berlin Heidelberg. 2007. P. 19–46.
36. **Pedersen M. E. H., Chipperfield A. J.** Parameter tuning versus adaptation: proof of principle study on differential evolution // *HL0803*. Hvas Laboratories. 2008.
37. **Smit S. K.** Parameter tuning and scientific testing in evolutionary algorithms. Vrije Universiteit, 2012.
38. **Mercer R. E., Sampson J. R.** Adaptive search using a reproductive meta-plan // *Kybernetes*. 1978. Vol. 7, N. 3. P. 215–228.
39. **Grefenstette J. J.** Optimization of control parameters for genetic algorithms // *IEEE Transactions on systems, man, and cybernetics*. 1986. Vol. 16, N. 1. P. 122–128.
40. **Hutter F., Hoos H. H., Stützle T.** Automatic algorithm configuration based on local search // *Aaai*. 2007. Vol. 7. P. 1152–1157.
41. **Yuan B., Gallagher M.** Combining Meta-EAs and racing for difficult EA parameter tuning tasks // *Parameter Setting in Evolutionary Algorithms*. Springer, Berlin, Heidelberg. 2007. P. 121–142.
42. **Hutter F., Hoos H. H., Leyton-Brown K.** Sequential model-based optimization for general algorithm configuration // *International Conference on Learning and Intelligent Optimization*. Springer, Berlin, Heidelberg. 2011. P. 507–523.

43. **Biau G.** Analysis of a random forests model // *Journal of Machine Learning Research*. 2012. Vol. 13, N. 1. P. 1063–1095.
44. **Myers R., Hancock E. R.** Empirical modelling of genetic algorithms // *Evolutionary computation*. 2001. Vol. 9, N. 4. P. 461–493.
45. **Taguchi G.** Taguchi methods: design of experiments. Vol. 4. Amer Supplier Inst, 1993.
46. **Adenso-Diaz B., Laguna M.** Fine-tuning of algorithms using fractional experimental designs and local search // *Operations Research*. 2006. Vol. 54, N. 1. P. 99–114.
47. **Pedersen M. E. H.** Tuning & simplifying heuristical optimization. Diss. University of Southampton, 2010.
48. **Goldsmann D., Nelson B. L., Schmeiser B.** Methods for selecting the best system // *Proceedings of the 23rd conference on Winter simulation*. IEEE Computer Society. 1991. P. 177–186.
49. **Branke J., Chick S. E., Schmidt C.** New developments in ranking and selection: an empirical comparison of the three main approaches // *Proceedings of the 37th conference on Winter simulation*. Winter Simulation Conference. IEEE. 2005. P. 708–717.
50. **Schmeiser B.** Simulation experiments. Vol. 2 // *Handbooks in operations research and management science*. 1990. P. 295–330.
51. **Rinott Y.** On two-stage selection procedures and related probability-inequalities // *Communications in Statistics-Theory and methods*. 1978. Vol. 7, N. 8. P. 799–811.
52. **Hochberg Y., Tamhane A. C.** Multiple comparison procedures. New York, NY: John Wiley & Sons, Inc, 1987.
53. **Kim S. H., Nelson B. L.** A fully sequential procedure for indifference-zone selection in simulation // *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. 2001. Vol. 11, N. 3. P. 251–273.
54. **Balaprakash P., Birattari M., Stützle T.** Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement // *International workshop on hybrid metaheuristics*. Springer, Berlin, Heidelberg. 2007. P. 108–122.
55. **Maron O., Moore A. W.** The racing algorithm: Model selection for lazy learners // *Lazy learning*. Springer, Dordrecht. 1997. P. 193–225.
56. **Birattari M., Kacprzyk J.** Tuning metaheuristics: a machine learning perspective. Berlin: Springer, 2009. Vol. 197. P. 85–114.
57. **Barros P. A. Jr., Kirby M. R., Mavris D. N.** Impact of sampling techniques selection on the creation of response surface models // *SAE transactions*. 2004. Vol. 113, N. 1. P. 1682–1693.
58. **Czarn A.** et al. Statistical exploratory analysis of genetic algorithms // *IEEE Transactions on Evolutionary Computation*. 2004. Vol. 8, N. 4. P. 405–421.
59. **Ramos I. C. O.** et al. Logistic regression for parameter tuning on an evolutionary algorithm // *IEEE Congress on Evolutionary Computation*. IEEE. 2005. Vol. 2. P. 1061–1068.
60. **Coy S. P.** et al. Using experimental design to find effective parameter settings for heuristics // *Journal of Heuristics*, Vol. 7, N. 1. P. 77–97.
61. **Bartz-Beielstein T., Parsopoulos K. E., Vrahatis M. N.** Analysis of particle swarm optimization using computational statistics // *International conference on numerical analysis and applied mathematics ICNAAM*. 2004. P. 34–37.
62. **Banzhaf W., Lasarczyk C.** Genetic programming of an algorithmic chemistry // In: *Genetic Programming Theory and Practice II*. Springer US, 2005. P. 175–190.
63. **Hutter F.** et al. Sequential model-based parameter optimization: An experimental investigation of automated and interactive approaches // *Experimental Methods for the Analysis of Optimization Algorithms*. Springer, Berlin, Heidelberg. 2010. P. 363–414.
64. **Ruxton G. D.** The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test // *Behavioral Ecology*. 2006. Vol. 17, N. 4. P. 688–690.
65. **Ugolotti R.** et al. GPU-based Automatic Configuration of Differential Evolution: a case study // *Portuguese Conference on Artificial Intelligence*. Springer, Berlin, Heidelberg. 2013. P. 114–125.
66. **Hutter F.** et al. ParamILS: an automatic algorithm configuration framework // *Journal of Artificial Intelligence Research*. 2009. Vol. 36, N. 1. P. 267–306.
67. **Nannen V., Eiben A. E.** A method for parameter calibration and relevance estimation in evolutionary algorithms // *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM. 2006. P. 183–190.
68. **Smit S. K., Eiben A. E.** Multi-problem parameter tuning using bonesa // *Artificial Evolution*. 2011. P. 222–233.
69. **Branke J., Schmidt C., Schmec H.** Efficient fitness estimation in noisy environments // *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc. 2001. P. 243–250.
70. **Zitzler E.** et al. SPEA2: Improving the strength Pareto evolutionary algorithm // *TIK-report*. 2001. Vol. 103. P. 95–100.
71. **Deb K.** et al. A fast and elitist multiobjective genetic algorithm: NSGA-II // *IEEE transactions on evolutionary computation*. 2002. Vol. 6, N. 2. P. 182–197.
72. **Deb K., Srinivasan A.** Innovization: Innovating design principles through optimization // *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM. Seattle, Washington, USA. 2006. P. 1629–1636.
73. **Branke J., Elomari J. A.** Meta-optimization for parameter tuning with a flexible computing budget // *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. ACM. 2012. P. 1245–1252.
74. **Fitzgerald T.** et al. ReACT: Real-Time Algorithm Configuration through Tournaments // *Seventh Annual Symposium on Combinatorial Search*. 2014.
75. **Powell M. J. D.** The BOBYQA algorithm for bound constrained optimization without derivatives // *Cambridge NA Report NA2009/06*, University of Cambridge, Cambridge. 2009. P. 26–46.
76. **Adams B. M.** et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, Sandia National Laboratories, Tech. Rep. SAND2010-2183, 2009.
77. **Csendes T., Pál L., Sendin J. O. H., Banga J. R.** The GLOBAL optimization method revisited // *Optimization Letters*. 2008. Vol. 2, N. 4. P. 445–454.
78. **Plantenga T. D.** Hopspack 2.0 user manual, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, SAND2009-6265, 2009.
79. **Powell M. J. D.** Developments of NEWUOA for minimization without derivatives // *IMA journal of numerical analysis*. 2008. Vol. 28, N. 4. P. 649–664.
80. **Le Digabel S., Tribes C.** NOMAD User Guide Version 3.5. Groupe d'études et de recherche en analyse des décisions. 2009. P. 1–45.
81. **Custódio A. L., Vicente L. N.** SID-PSM: A Pattern Search Method Guided by Simplex Derivatives for Use in Derivative-Free Optimization // *Departamento de Matemática, Universidade de Coimbra, Coimbra*. 2008.
82. **Optimization Algorithms [Электронный ресурс]** // modeFRONTIER by Esteco. URL: <http://www.esteco.com/mode-frontier/optimization-algorithms> (дата обращения: 25.01.2018).
83. **Chase N.** et al. A benchmark study of optimization search algorithms // *Red Cedar Technology*. 2010.
84. **Generic Tool for Optimization, GT Opt** // Datadvance. URL: <https://www.datadvance.net/product/pseven-core/generic-tool-for-optimization/> (дата обращения: 25.01.2018).
85. **Egorov I. N.** et al. IOSO optimization toolkit-novel software to create better design // *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2002. P. 4–6.
86. **Hutter F., Ramage S.** Manual for SMAC version v2. 10.02-master. Vancouver: Department of Computer Science University of British Columbia, 2015.

87. **Blot A.** et al. MO-ParamILS: A Multi-objective Automatic Algorithm Configuration Framework // International Conference on Learning and Intelligent Optimization. Springer International Publishing, 2016. P. 32–47.

88. **Ansótegui C., Sellmann M., Tierney K.** A gender-based genetic algorithm for the automatic configuration of algorithms // International Conference on Principles and Practice of Constraint Programming. Springer, Berlin, Heidelberg, 2009. P. 142–157.

89. **López-Ibáñez M.** et al. The irace package: Iterated racing for automatic algorithm configuration // IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.

90. **López-Ibáñez M., Stutzle T.** The automatic design of multiobjective ant colony optimization algorithms // IEEE Transactions on Evolutionary Computation. 2012. Vol. 16, N. 6. P. 861–875.

91. **Dubois-Lacoste J., López-Ibáñez M., Stützle T.** Automatic configuration of state-of-the-art multi-objective optimizers using the TP + PLS framework // Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM, 2011. P. 2019–2026.

92. **Battiti R., Passerini A.** Brain-computer evolutionary multiobjective optimization (BC-EMO): a genetic algorithm adapting to the decision maker // IEEE Transaction on Evolutionary Computation. 2010. Vol. 14, N. 5. P. 671–687.

93. **Карпенко А. П., Моор Д. А., Мухлисуллина Д. Т.** Многокритериальная оптимизация на основе нейронечеткой аппроксимации функции предпочтений лица, принимающего решения // Наука и образование. 2010. № 6.

**T. A. Agasiev**, Ph.D. student, Assistant, email: agtaleh@mail.ru,

**A. P. Karpenko**, Doctor of Sc., Professor, email: apkarpenko@mail.ru

Bauman Moscow State Technical University, Moscow, 105005, Russian Federation

## Modern Techniques of Global Optimization. Review

*A global constrained optimization problem with high computational complexity of an objective function was formulated. The following terms have been defined: optimization problem features, base problem and base algorithm, meta-problem and meta-algorithm, base problem strategy, indicator of the strategy efficiency. Each optimization problem can be represented by a vector of its features, describing those properties of the problem and objective function that are meaningful in terms of optimization algorithms efficiency. The vector of problems features, estimated by means of exploratory landscape analysis methods, can be used to classify optimization problems or to gain important information about the problem definition in order to find an appropriate strategy of optimization algorithm to be used. To compare candidate strategies and select arguably the most suitable one the efficiency indicator should be determined. The article considers various ways of the estimation of optimization algorithms efficiency. The problem of finding the best algorithm's strategies based on a vector of problem's features and a given efficiency indicator is called meta-optimization problem. Statements of multi-indicator, multi-class and multi-budget meta-optimization problems are provided. The result of solving the meta-optimization problem is a set of the most appropriate strategies, i.e. the optimization algorithm tuned for solving particular types of optimization problems. Different approaches to automated tuning of optimization algorithms are described. A review of advanced exploratory landscape analysis and meta-optimization methods was performed based on approximately 100 articles. The article concludes with considering a modern software for solving the base and meta-optimization problems.*

**Keywords:** global optimization, exploratory landscape analysis, meta-optimization, surrogate modelling, performance analysis of optimization algorithms

### References

1. **Shan S., Wang G. G.** Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, *Structural and Multidisciplinary Optimization*, 2010, vol. 41, no. 2, pp. 219–241.

2. **Wolpert D. H., Macready W. G.** No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, 1997, vol. 1, no. 1, pp. 67–82.

3. **Rios L. M., Sahinidis N. V.** Derivative-free optimization: a review of algorithms and comparison of software implementations, *Journal of Global Optimization*, 2013, vol. 56, no. 3, pp. 1247–1293.

4. **Kirkpatrick S.** Optimization by simulated annealing: Quantitative studies, *Journal of Statistical Physics*, 1984, vol. 34, no. 5–6, pp. 975–986.

5. **Liepins G. E., Hilliard M. R.** Genetic algorithms: Foundations and applications, *Annals of Operations Research*, 1989, vol. 21, no. 1, pp. 31–58.

6. **Hansen N.** The CMA evolution strategy: a comparing review, *Towards a new evolutionary computation*, Springer Berlin Heidelberg, 2006, pp. 75–102.

7. **Poli R., Kennedy J., Blackwell T.** Particle swarm optimization, *Swarm intelligence*, 2007, vol. 1, no. 1, pp. 33–57.

8. **Karaboga D., Basturk B.** A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 2007, vol. 39, no. 3, pp. 459–471.

9. **Dorigo M., Birattari M., Stutzle T.** Ant colony optimization, *IEEE Computational Intelligence Magazine*, 2006, vol. 1, no. 4, pp. 28–39.

10. **Jones D. R., Schonlau M., Welch W. J.** Efficient global optimization of expensive black-box functions, *Journal of Global Optimization*, 1998, vol. 13, no. 4, pp. 455–492.

11. **Cox D. D., John S.** SDO: A Statistical Method for Global Optimization, *Multidisciplinary Design Optimization: State of the Art*, 1997, vol. 80, pp. 315–329.

12. **Huyer W., Neumaier A.** SNOBFIT — Stable noisy optimization by branch and fit, *ACM Transactions on Mathematical Software (TOMS)*, 2008, vol. 35, no. 2, pp. 1–21.

13. **Hansen N.** et al. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Diss. INRIA, 2010.



14. **Decison** Tree for Optimization Software, available at: <http://plato.asu.edu/guide.html> (accessed 25.01.2018).
15. **Momin J., Yang X. S.** A literature survey of benchmark functions for global optimization problems, *Journal of Mathematical Modelling and Numerical Optimisation*, 2013, vol. 4, no. 2, pp. 150–194.
16. **Gould N. I. M., Orban D., Toint P. L.** CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications*, 2015, vol. 60, no. 3, pp. 545–557.
17. **Global** Optimization Test Problems, available at: <http://www.mat.univie.ac.at/~neum/glopt/test.html> (accessed 25.01.2018).
18. **Dolan E. D., Moré J. J., Munson T. S.** Benchmarking optimization software with COPS 3.0, *Argonne National Lab*, IL (US), 2004, no. ANL/MCS-TM-273.
19. **Floudas C. A.** et al. Handbook of test problems in local and global optimization, Springer Science & Business Media, 2013, vol. 33.
20. **Beiranvand V., Hare W., Lucet Y.** Best practices for comparing optimization algorithms, *Optimization and Engineering*, 2017, vol. 18, no. 4, pp. 815–848.
21. **Dolan E. D., Moré J. J.** Benchmarking optimization software with performance profiles, *Mathematical Programming*, vol. 91, no. 2, pp. 201–213.
22. **Gould N., Scott J.** A Note on Performance Profiles for Benchmarking Software, *ACM Transactions on Mathematical Software (TOMS)*, 2016, vol. 43, no. 2, p. 15.
23. **Moré J. J., Wild S. M.** Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization*, 2009, vol. 20, no. 1, pp. 172–191.
24. **Bongartz I.** et al. CUTE: Constrained and unconstrained testing environment, *ACM Transactions on Mathematical Software (TOMS)*, 1995, vol. 21, no. 1, pp. 123–160.
25. **Mersmann O.** et al. Exploratory landscape analysis, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 829–836.
26. **Ugolotti R.** Meta-optimization of Bio-inspired Techniques for Object Recognition. Diss. Università di Parma. Dipartimento di Ingegneria dell'Informazione, 2015.
27. **Smit S. K., Eiben A. E., Szilávik Z.** An MOEA-based Method to Tune EA Parameters on Multiple Objective Functions, *IJCCI (ICEC)*, 2010, pp. 261–268.
28. **Li G., Aute V., Azarm S.** An accumulative error based adaptive design of experiments for offline metamodeling, *Structural and Multidisciplinary Optimization*, 2010, vol. 40, no. 1, pp. 137–155.
29. **Kerschke P.** et al. Cell mapping techniques for exploratory landscape analysis, *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*. Springer, Cham., 2014, pp. 115–131.
30. **Flamm C.** et al. Barrier trees of degenerate landscapes, *Zeitschrift für Physikalische Chemie*, 2002, vol. 216, no. 2, p. 155.
31. **Muñoz M. A., Kirley M., Halgamuge S. K.** Exploratory landscape analysis of continuous space optimization problems using information content, *IEEE Transactions on Evolutionary Computation*, 2015, vol. 19, no. 1, pp. 74–87.
32. **Preuss M.** Improved Topological Nicheing for Real-Valued Global Optimization, *European Conference on the Applications of Evolutionary Computation*, Springer, Berlin, Heidelberg, 2012, pp. 386–395.
33. **Kerschke P.** et al. Detecting funnel structures by means of exploratory landscape analysis, *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 265–272.
34. **Munoz M. A., Smith-Miles K.** Effects of function translation and dimensionality reduction on landscape analysis, *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1336–1342.
35. **Eiben A. E., Michalewicz Z., Schoenauer M., Smith J. E.** Parameter Control in Evolutionary Algorithms, *Parameter Setting in Evolutionary Algorithms*. Springer Berlin Heidelberg, 2007, pp. 19–46.
36. **Pedersen M. E. H., Chipperfield A. J.** Parameter tuning versus adaptation: proof of principle study on differential evolution, HL0803. Hvass Laboratories, 2008.
37. **Smit S. K.** Parameter tuning and scientific testing in evolutionary algorithms, Vrije Universiteit, 2012.
38. **Mercer R. E., Sampson J. R.** Adaptive search using a re-productive meta-plan, *Kybernetes*, 1978, vol. 7, no. 3, pp. 215–228.
39. **Grefenstette J. J.** Optimization of control parameters for genetic algorithms, *IEEE Transactions on systems, man, and cybernetics*, 1986, v.ol. 16, no. 1, pp. 122–128.
40. **Hutter F., Hoos H. H., Stützle T.** Automatic algorithm configuration based on local search, *Aaai*, 2007, vol. 7, pp. 1152–1157.
41. **Yuan B., Gallagher M.** Combining Meta-EAs and racing for difficult EA parameter tuning tasks, *Parameter Setting in Evolutionary Algorithms*, Springer, Berlin, Heidelberg, 2007, pp. 121–142.
42. **Hutter F., Hoos H. H., Leyton-Brown K.** Sequential model-based optimization for general algorithm configuration, *International Conference on Learning and Intelligent Optimization*, Springer, Berlin, Heidelberg, 2011, pp. 507–523.
43. **Biau G.** Analysis of a random forests model, *Journal of Machine Learning Research.*, 2012, vol. 13, no. 1, pp. 1063–1095.
44. **Myers R., Hancock E. R.** Empirical modelling of genetic algorithms, *Evolutionary Computation*, 2001, vol. 9, no. 4, pp. 461–493.
45. **Taguchi G.** Taguchi methods: design of experiments, vol. 4. Amer Supplier Inst, 1993.
46. **Adenso-Diaz B., Laguna M.** Fine-tuning of algorithms using fractional experimental designs and local search, *Operations Research*, 2006, vol. 54, no. 1, pp. 99–114.
47. **Pedersen M. E. H.** Tuning & simplifying heuristical optimization, Diss. University of Southampton, 2010.
48. **Goldman D., Nelson B. L., Schmeiser B.** Methods for selecting the best system, *Proceedings of the 23rd conference on Winter simulation*. IEEE Computer Society, 1991, pp. 177–186.
49. **Branke J., Chick S. E., Schmidt C.** New developments in ranking and selection: an empirical comparison of the three main approaches, *Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, IEEE*, 2005, pp. 708–717.
50. **Schmeiser B.** Simulation experiments. Vol. 2, *Handbooks in Operations Research and Management Science*, 1990, pp. 295–330.
51. **Rinott Y.** On two-stage selection procedures and related probability-inequalities, *Communications in Statistics-Theory and methods*, 1978, vol. 7, no. 8, pp. 799–811.
52. **Hochberg Y., Tamhane A. C.** Multiple comparison procedures. New York, NY, John Wiley & Sons, Inc, 1987.
53. **Kim S. H., Nelson B. L.** A fully sequential procedure for indifference-zone selection in simulation, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2001, vol. 11, no. 3, pp. 251–273.
54. **Balaprakash P., Birattari M., Stützle T.** Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement, *International workshop on hybrid metaheuristics*, Springer, Berlin, Heidelberg, 2007, pp. 108–122.
55. **Maron O., Moore A. W.** The racing algorithm: Model selection for lazy learners. *Lazy learning*, Springer, Dordrecht, 1997, pp. 193–225.
56. **Birattari M., Kacprzyk J.** Tuning metaheuristics: a machine learning perspective, Berlin, Springer, 2009, vol. 197, pp. 85–114.
57. **Barros P. A. Jr., Kirby M. R., Mavris D. N.** Impact of sampling techniques selection on the creation of response surface models, *SAE transactions*, 2004, vol. 113, no. 1, pp. 1682–1693.
58. **Czarn A.** et al. Statistical exploratory analysis of genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 2004, vol. 8, no. 4, pp. 405–421.
59. **Ramos I. C. O.** et al. Logistic regression for parameter tuning on an evolutionary algorithm, *IEEE Congress on Evolutionary Computation*. IEEE, 2005, vol. 2, pp. 1061–1068.
60. **Coy S. P.** et al. Using experimental design to find effective parameter settings for heuristics, *Journal of Heuristics*, vol. 7, no. 1, pp. 77–97.

61. **Bartz-Beielstein T., Parsopoulos K. E., Vrahatis M. N.** Analysis of particle swarm optimization using computational statistics, *International conference on numerical analysis and applied mathematics ICNAAM*, 2004, pp. 34–37.
62. **Banzhaf W., Lasarczyk C.** Genetic programming of an algorithmic chemistry, *Genetic Programming Theory and Practice II*. Springer US, 2005, pp. 175–190.
63. **Hutter F.** et al. Sequential model-based parameter optimization: An experimental investigation of automated and interactive approaches, *Experimental Methods for the Analysis of Optimization Algorithms*, Springer, Berlin, Heidelberg, 2010, pp. 363–414.
64. **Ruxton G. D.** The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test, *Behavioral Ecology*, 2006, vol. 17, no. 4, pp. 688–690.
65. **Ugolotti R.** et al. GPU-based Automatic Configuration of Differential Evolution: a case study, *Portuguese Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, 2013, pp. 114–125.
66. **Hutter F.** et al. ParamILS: an automatic algorithm configuration framework, *Journal of Artificial Intelligence Research*, 2009, vol. 36, no. 1, pp. 267–306.
67. **Nannen V., Eiben A. E.** A method for parameter calibration and relevance estimation in evolutionary algorithms, *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM*, 2006, pp. 183–190.
68. **Smit S. K., Eiben A. E.** Multi-problem parameter tuning using bonesa, *Artificial Evolution*, 2011, pp. 222–233.
69. **Branke J., Schmidt C., Schmeck H.** Efficient fitness estimation in noisy environments, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc., 2001, pp. 243–250.
70. **Zitzler E.** et al. SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-report*, 2001, vol. 103, pp. 95–100.
71. **Deb K.** et al. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE transactions on evolutionary computation*, 2002, vol. 6, no. 2, pp. 182–197.
72. **Deb K., Srinivasan A.** Innovization: Innovating design principles through optimization, *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM*. Seattle, Washington, USA, 2006, pp. 1629–1636.
73. **Branke J., Elomari J. A.** Meta-optimization for parameter tuning with a flexible computing budget, *Proceedings of the 14th annual conference on Genetic and evolutionary computation. ACM*, 2012, pp. 1245–1252.
74. **Fitzgerald T.** et al. ReACT: Real-Time Algorithm Configuration through Tournaments, *Seventh Annual Symposium on Combinatorial Search*, 2014.
75. **Powell M. J. D.** The BOBYQA algorithm for bound constrained optimization without derivatives, *Cambridge NA Report NA2009/06*, University of Cambridge, Cambridge, 2009, pp. 26–46.
76. **Adams B. M.** et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, Sandia National Laboratories, Tech. Rep. SAND2010-2183, 2009.
77. **Csendes T., Pál L., Sendín J. O. H., Banga J. R.** The GLOBAL optimization method revisited, *Optimization Letters*, 2008, vol. 2, no. 4, pp. 445–454.
78. **Plantenga T. D.** Hopspack 2.0 user manual, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, SAND2009-6265, 2009.
79. **Powell M. J. D.** Developments of NEWUOA for minimization without derivatives, *IMA Journal of Numerical Analysis*, 2008, vol. 28, no. 4, pp. 649–664.
80. **Le Digabel S., Tribes C.** NOMAD User Guide Version 3.5. Groupe d'études et de recherche en analyse des décisions, 2009, pp. 1–45.
81. **Custódio A. L., Vicente L. N.** SID-PSM: A Pattern Search Method Guided by Simplex Derivatives for Use in Derivative-Free Optimization, *Departamento de Matemática, Universidade de Coimbra*, Coimbra, 2008.
82. **Optimization Algorithms (modeFRONTIER by Esteco)**, available at: <http://www.esteco.com/modefrontier/optimization-algorithms> (accessed 25.01.2018).
83. **Chase N.** et al. A benchmark study of optimization search algorithms, Red Cedar Technology, 2010.
84. **Generic Tool for Optimization, GT Opt (Datadvance)**, available at: <https://www.datadvance.net/product/pseven-core/generic-tool-for-optimization/> (accessed 25.01.2018).
85. **Egorov I. N.** et al. IOSO optimization toolkit-novel software to create better design, *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, pp. 4–6.
86. **Hutter F., Ramage S.** Manual for SMAC version v2.10.02-master. Vancouver: Department of Computer Science University of British Columbia, 2015.
87. **Blot A.** et al. MO-ParamILS: A Multi-Objective Automatic Algorithm Configuration Framework, *International Conference on Learning and Intelligent Optimization*, Springer International Publishing, 2016, pp. 32–47.
88. **Ansótegui C., Sellmann M., Tierney K.** A gender-based genetic algorithm for the automatic configuration of algorithms, *International Conference on Principles and Practice of Constraint Programming*, Springer, Berlin, Heidelberg, 2009, pp. 142–157.
89. **López-Ibáñez M.** et al. The irace package: Iterated racing for automatic algorithm configuration, IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.
90. **López-Ibáñez M., Stützle T.** The automatic design of multiobjective ant colony optimization algorithms, *IEEE Transactions on Evolutionary Computation*, 2012, vol. 16, no. 6, pp. 861–875.
91. **Dubois-Lacoste J., López-Ibáñez M., Stützle T.** Automatic configuration of state-of-the-art multi-objective optimizers using the TP + PLS framework, *Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM*, 2011, pp. 2019–2026.
92. **Battiti R., Passerini A.** Brain-computer evolutionary multiobjective optimization (BC-EMO): a genetic algorithm adapting to the decision maker, *IEEE Transaction on Evolutionary Computation*, 2010, vol. 14, no. 5, pp. 671–687.
93. **Karpenko A. P., Moor D. A., Mukhlisullina D. T.** *Mnogokriterialnaja optimizacija na osnove nejro-nechetkoj aproksimacii funkcii predpochenij lica, prinimajushhego reshenija* (Multicriteria optimization based on neuro-fuzzy approximation of decision maker's utility function). *Science and Education of Bauman MSTU*, 2010, no. 6 (in Russian).

**Е. М. Бронштейн**, д-р физ.-мат. наук, проф., e-mail: bro-efim@yandex.ru,  
Уфимский государственный авиационный технический университет,  
**Р. В. Гиндуллин**, канд. физ.-мат. наук, e-mail: gramiz@mail.ru,  
Башкирский государственный университет

## Решение задачи о заборе и доставке грузов одним транспортным средством с использованием пакета CPLEX

*Рассматривается задача формирования цикла минимальной длины, обеспечивающего доставку грузов разных весов от производителей потребителям при условии доставки груза от каждого производителя конкретному потребителю одним транспортным средством ограниченной вместимости. Рассмотрен частный случай задачи, в котором каждый груз должен быть сразу доставлен потребителю. Проведено сравнение эффективности построенных формализаций для общего и частного случаев задачи.*

**Ключевые слова:** маршрутизация, цикл, оптимизация, линейное целочисленное программирование

### Введение

Впервые задача маршрутизации транспортных средств (VRP — Vehicle Routing Problem) была поставлена в работе [1]. За истекшие десятилетия рассматривалось множество модификаций VRP. Информация по этой тематике аккумулируется на сайте [2].

В статье рассматривается вариант задачи VRP, именуемый "The pickup and delivery problem" (PDP) с одним транспортным средством ограниченной вместимости (SPDP), в котором предполагается наличие полного графа, вершинами которого являются пункты производства и потребления, а дугами — соответствующие пути, и единственного транспортного средства ограниченной вместимости, которое должно доставить грузы различного веса от поставщиков (производителей) потребителям, причем груз от каждого производителя должен быть доставлен конкретному потребителю. Задача относится к классу NP-сложных задач. Задачи вида SPDP также называют Pickup-Delivery Traveling Salesman Problem или Traveling Salesman Problem with Pickups and Deliveries [3].

Задача SPDP без учета вместимости транспортных средств и, соответственно, весов грузов была решена точным способом в работе [4]. Пример с 15 пунктами решался комбинацией методов ветвей и отрезков и жадного поиска.

В работах [5, 6] использовались различные подходы с перестановками пунктов в циклах для решения задачи SPDP без учета вместимости транспортных средств и весов грузов. Подходы были опробованы на группе из 108 задач из TSPLIB, включая задачу с 441 пунктом.

Также рассматривались родственные задачи:

- *Pickup and delivery problem with time windows* — вариант рассматриваемой задачи, в котором добавлены ограничения на временные окна. Успешно использовались методы ветвей и границ или ветвей и отсечений (генерации столбцов) [7]. В работе [8] данный подход развивается для задачи с временными окнами и несколькими транспортными средствами. Модели с полиномиально растущим числом ограничений там названы компактными;
- *задача "доставки по звонку"* (Dial-a-ride problem, DARP): имеются одно или несколько транспортных средств, а клиенты указывают пункты отправки и доставки людей или скоропортящихся грузов, что накладывает ограничения на время перевозки данных грузов. В качестве целевой функции вместо минимальных расстояний используется минимальное время ожидания клиента. В работе [9] приведен обзор методов решения задач данной категории.

В работе [10] построен ряд формализаций задачи SPDP с ограниченной вместимостью, в частности адаптацией описанных в обзорах [11, 12] приемов, применявшихся к квадратичной задаче о назначениях [13].

### 1. Постановка задачи

Пусть  $P = \{1, \dots, n\}$  — пункты вывоза грузов;  $D = \{n + 1, \dots, 2n\}$  — пункты доставки грузов; вес груза в  $i$ -м пункте  $q_i$  ( $i = \overline{1, n}$ ). Множество пунктов есть  $V = P \cup D \cup \{0\}$ , где нулевой пункт является базой. Транспортное средство (ТС)

вместимостью  $S$  должно объехать все пункты по циклу таким образом, чтобы доставить грузы из  $i$ -го пункта в  $(n + i)$ -й при всех  $i$ . Полагаем, что в  $(n + i)$ -м пункте вес груза отрицательный ( $-q_i$ ). Маршрут должен начинаться и заканчиваться в базовом пункте. Известны расстояния между всеми парами пунктов  $c_{ij}$ . Требуется найти допустимый цикл минимальной длины. Задача относится к классу NP-трудных, поскольку в случае, когда пункты каждой пары совпадают и вместимость ТС не ограничена, получим классическую задачу коммивояжера.

Сформулируем несколько свойств данной задачи.

1. Минимально допустимая вместимость ТС равна  $\max\{q_i\}$ . Действительно, очевидно, что при  $S < \max\{q_i\}$  организовать перевозку невозможно. При  $S = \max\{q_i\}$  допустимым является маршрут  $0-1-(n+1)-2-(n+2)-\dots-n-2n-0$ . Тем самым, эта задача существенно отличается от более общей задачи транспортировки однородного груза, для которой задача вычисления минимально допустимой вместимости ТС является NP-трудной [10].

2. При неограниченной вместимости ТС число допустимых маршрутов равно  $(2n)!/2^n$ . Действительно, всего перестановок пунктов  $(2n)!$ , при этом каждая из допустимых перестановок порождает  $2^n$  перестановок, полученных всевозможными перестановками пар пунктов с номерами  $i, (n + i)$  при  $i = 1, 2, \dots, n$ . Этот результат получен в работе [4] более сложным рассуждением.

3. При  $S \geq \max\{q_i\}$  любой допустимый отрезок маршрута можно продолжить. Действительно, если после прохождения отрезка ТС не содержит груза и при этом есть необслуженные пункты, то в качестве следующего можно принять любой пункт, в котором есть груз. Если ТС содержит груз, то это означает, что его забрали в некотором пункте  $i$ , но в  $(n + i)$ -й не доставили. Следующим пунктом маршрута можно принять  $(n + i)$ -й.

В работе [10] приведены одна квадратичная и восемь линейных целочисленных формализаций рассматриваемой задачи. Проведенные численные эксперименты показали, что наиболее эффективными моделями из работы [10] оказались двух- и трехиндексная целочисленные линейные модели. Приведем эти модели. Следует отметить, что в предложенных моделях число ограничений растет полиномиально с ростом размерности. Именно это позволяет использовать для решения задачи стандартные программные продукты.

*Двухиндексная линейная модель.* Пусть  $t_{ij}$  — булевы переменные, равные 1, если в цикле дуга из пункта  $i$  ведет в пункт  $j$ .

Ограничения:

$$\sum_{i=0}^{2n} t_{ij} = 1 \quad (j = 0, 1, \dots, 2n); \quad (1)$$

$$\sum_{j=0}^{2n} t_{ij} = 1 \quad (i = 0, 1, \dots, 2n); \quad (2)$$

$$t_{ii} = 0 \quad (i = 0, \dots, 2n). \quad (3)$$

Для того чтобы избежать подциклов, следуя работе [11], вводятся вещественные переменные  $A_i$ , удовлетворяющие следующим ограничениям:

$$0 \leq A_i \leq 2n \quad (i = 1, \dots, 2n), \quad A_0 = 0; \quad (4)$$

$$A_i - A_j + (2n + 1)t_{ij} \leq 2n \quad (i, j = 1, \dots, 2n). \quad (5)$$

Из условий (4), (5) следует, что величины  $A_i$  автоматически целые, равные номерам пунктов в порядке прохождения в цикле.

Ограничение на правильность прохождения цикла:

$$A_i \leq A_{n+i} \quad (i = 1, \dots, 2). \quad (6)$$

Ограничение на вместимость транспортного средства:

$$\sum_{i=0}^{2n} q_i [A_i \leq k] \leq S \quad (k = 2, \dots, 2n - 2). \quad (7)$$

Здесь  $[a]$  — численное значение логической величины  $a$ .

Целевая функция:

$$\sum_{i=0}^{2n} \sum_{j=0}^{2n} c_{ij} t_{ij} \rightarrow \min. \quad (8)$$

*Трехиндексная линейная модель.* Пусть  $v_{ij}^k$  ( $i, j = 0, \dots, 2n; k = 1, \dots, 2n + 1$ ) — булевы переменные, равные 1, если  $k$ -я по порядку дуга в маршруте ведет из пункта  $i$  в пункт  $j$ . Ограничения:

$$\sum_{i=0}^{2n} \sum_{k=0}^{2n+1} v_{ij}^k = 1 \quad (j = 0, \dots, 2n); \quad (9)$$

$$\sum_{j=0}^{2n} \sum_{k=0}^{2n+1} v_{ij}^k = 1 \quad (i = 0, \dots, 2n); \quad (10)$$

$$\sum_{i=0}^{2n} \sum_{j=0}^{2n} v_{ij}^k = 1 \quad (k = 0, \dots, 2n + 1); \quad (11)$$

$$\sum_{j=0}^n v_{0j}^1 = 1, \quad \sum_{i=n+1}^{2n} v_{i0}^{2n+1} = 1. \quad (12)$$

Условие последовательного прохождения дуг (конец дуги  $k$  совпадает с началом дуги  $k + 1$ ):

$$\sum_{i=0}^{2n} v_{ij}^k = \sum_{s=0}^{2n} v_{js}^{k+1} \quad (j = 1, \dots, 2n; k = 1, \dots, 2n). \quad (13)$$

Условия правильности прохождения пунктов:

$$\sum_{s=1}^{2n} \sum_{k \leq s} \sum_{p=1}^{2n} \sum_{j=1}^{2n} (v_{(n+i)j}^k - v_{ip}^k) \leq 0 \quad (i = 1, \dots, n). \quad (14)$$

Действительно, если из  $i$ -го пункта ТС выезжает по дуге с номером  $k_1$ , а из  $(n + i)$ -го — по дуге с номером  $k_2$ , то  $\sum_{p=1}^{2n} \sum_{j=1}^{2n} (v_{(n+i)p}^k - v_{ij}^k) = 0$  при  $k \leq \min(k_1, k_2)$ , поскольку в этом случае  $v_{(n+i)p}^k = v_{ij}^k = 0$  при любых  $p, i$ , т. е.  $\sum_{k \leq s} \sum_{p=1}^{2n} \sum_{j=1}^{2n} (v_{(n+i)p}^k - v_{ip}^k) = 0$ , если  $s \leq \min(k_1, k_2)$ . Это же равенство справедливо при  $s \geq \max(k_1, k_2)$ , поскольку в этом случае  $v_{(n+i)p}^{k_1} = v_{ij}^{k_2} = 1$  при некоторых однозначно определенных  $p, j, k_1, k_2$  и  $v_{(n+i)p}^k = v_{ij}^k = 0$  при всех остальных  $p, j, k$ . Если  $k_1 < k_2$ ,  $v_{ip}^{k_1} = 1$  для некоторого  $p$  и  $v_{ip}^k = 0$  при остальных  $p$ ,  $v_{(n+i)p}^k = 0$  для всех  $p$ , т. е. неравенство (14) выполняется. Аналогично проверяется, что при  $k_2 < k_1$  неравенство (14) не выполняется.

Ограничение на вместимость:

$$\sum_{k \leq s} \sum_{i=1}^{2n} \sum_{j=1}^{2n} v_{ij}^k q_i \leq S \quad (s = 1, \dots, 2n). \quad (15)$$

Целевая функция:

$$\sum_{k=1}^{2n+1} \sum_{i=1}^{2n} \sum_{j=1}^{2n} v_{ij}^k c_{ij} \rightarrow \min. \quad (16)$$

Избыточные ограничения обычно ускоряют решение задачи. В работе [10] предложены два вида дополнительных ограничений для предложенных моделей.

*Дополнительные ограничения 1-го вида.* Сумма грузов в двух следующих друг за другом пунктах производства не должна превышать вместимости ТС. Аналогично для пунктов потребления.

• Для модели (1)—(8) ограничения имеют вид:

$$t_{ij} \leq [q_i + q_j \leq S] \quad (i, j = 1, \dots, n); \quad (17)$$

$$t_{(i+n)(j+n)} \leq [q_i + q_j \leq S] \quad (i, j = 1, \dots, n); \quad (18)$$

• для модели (9)—(16):

$$v_{ij}^k \leq [q_i + q_j \leq S] \quad (19)$$

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

$$v_{(i+n)(j+n)}^k \leq [q_i + q_j \leq S] \quad (20)$$

$$(i, j = 1, \dots, n; k = 1, \dots, 2n + 1).$$

*Дополнительные ограничения 2-го вида.* Являются адаптацией ограничений для LIFO-случая [12]. Дуги, соединяющие пункты  $i - (n + j)$  и  $j - (n + i)$  при различных направлениях движения, одновременно в маршрут перевозок входить не могут, поэтому:

• для модели (1)—(8) добавляются ограничения:

$$t_{i(j+n)} + t_{j(i+n)} \leq 1 \quad (i, j = 1, \dots, n); \quad (21)$$

$$t_{i(j+n)} + t_{(i+n)j} \leq 1 \quad (i, j = 1, \dots, n); \quad (22)$$

$$t_{(i+n)j} + t_{(j+n)i} \leq 1 \quad (i, j = 1, \dots, n); \quad (23)$$

• для модели (9)—(16) — ограничения:

$$v_{i(j+n)}^k + v_{j(i+n)}^p \leq 1 \quad (24)$$

$$(i, j = 1, \dots, n; k, p = 1, \dots, 2n + 1);$$

$$v_{i(j+n)}^k + v_{(i+n)j}^p \leq 1 \quad (25)$$

$$(i, j = 1, \dots, n; k, p = 1, \dots, 2n + 1);$$

$$v_{(i+n)j}^k + v_{(j+n)i}^p \leq 1 \quad (26)$$

$$(i, j = 1, \dots, n; k, p = 1, \dots, 2n + 1).$$

*Частный случай.* В некоторых случаях важно, чтобы грузы из пунктов производства доставлялись в соответствующие пункты назначения без заезда в другие пункты. Это означает, что  $(n + i)$ -й пункт в цикле должен следовать непосредственно за  $i$ -м для любого  $i = 1, \dots, n$ . Данная задача может быть сведена к задаче коммивояжера с асимметричной матрицей расстояний.

В этом случае:

• в обеих моделях принимаем  $S = q_i = 1$

$$(i = 1, \dots, n);$$

• удаляются ограничения на вместимость;

• для модели (1)—(8) добавляются условия

$$t_{i, (n+i)} = 1 \quad (i = 1, \dots, n); \quad (27)$$

• для модели (9)—(16) добавляются условия

$$\sum_{k=1}^{2n+1} v_{i, (i+n)}^k = 1 \quad (i = 1, \dots, n). \quad (28)$$

## 2. Вычислительные эксперименты

В данном разделе приводятся результаты вычислительных экспериментов для оценки эффективности двух- и трехиндексной линейных моделей. При проведении вычислительного эксперимента использовался ПК со следующими характеристиками: процессор *Intel Celeron 1.6GHz 4Gb RAM*, операционная система *Windows 7 32bit*. Использована среда *OPL Cplex 12.6.1* как решатель задач линейного целочисленного программирования. Память, отведенная для расчетов, 2048 Мбайт.

Вычислительные эксперименты для моделей ставились на евклидовых задачах со случайно сгенерированными целочисленными координатами пунктов (включая базу) и весами  $q_i$ . Координаты пунктов для каждой точки вывоза или доставки генерировались случайным образом в соответствии с равномерным распределением на квадрате  $[0,100] \times [0,100]$ . Целочисленный вес каждого пункта  $q_i$  генерировался случайным образом в диапазоне  $[1,10]$ . Вместимость ТС принималась минимально допустимой, т.е. равной  $\max q_i$ .

Серии численных экспериментов были проведены для двух- и трехиндексных линейных моделей.

Для  $n = 3, 4, 5$  было сгенерировано по 50 примеров. Из-за высокой трудоемкости, при  $n = 6$  число экспериментов было сокращено до 20, при  $n = 7, 8$  — до 5. Среднее число итераций и средняя продолжительность нахождения решения примера в секундах представлены в табл. 1 и 2 соответственно. Если хоть один пример не был решен, то в поле ставится прочерк либо указывается число решенных примеров из всей совокупности сгенерированных примеров. Строка в таблице с наилучшим результатом выделена курсивом.

В ряде случаев повышение размерности ограничено памятью ПК, например, при  $n = 8$  только трехиндексная модель может решить сгенерированные примеры без переполнения выделенного объема памяти. Кроме того, стоит отметить, что так как модели являются точными, то решения во всех примерах совпадают. Проверка корректности полученных решений моделей была проведена сравнением с результатами полного перебора при  $n = 3$ .

Далее проведен анализ эффективности введения дополнительных ограничений в наиболее эффективной трехиндексной линейной модели.

Как и в общем случае, решались евклидовы задачи со случайно сгенерированными цело-

Таблица 1

Сравнение моделей (число итераций)

Число примеров	50	50	50	20	5	2
Размерность задачи $n$	3	4	5	6	7	8
Двухиндексная модель	690	$3,7 \cdot 10^4$	$9,0 \cdot 10^5$	$3,2 \cdot 10^7$	$4,0 \cdot 10^7$ (решено 4/5 примеров)	—
Трехиндексная модель	<i>8</i>	<i>418</i>	<i><math>1,2 \cdot 10^4</math></i>	<i><math>9,5 \cdot 10^4</math></i>	<i><math>4,3 \cdot 10^5</math></i>	<i><math>1,3 \cdot 10^7</math></i>

Таблица 2

Сравнение моделей (с)

Число примеров	50	50	50	20	5	5
Размерность задачи $n$	3	4	5	6	7	8
Двухиндексная модель	0,1	1,8	53,6	1623,9	2855 (решено 4/5 примеров)	—
Трехиндексная модель	<i>0,0</i>	<i>0,4</i>	<i>1,7</i>	<i>16,6</i>	<i>134,6</i>	<i>3390,9</i>

Таблица 3

Сравнение эффективности ввода в модели дополнительных ограничений (число итераций)

$n$	Трехиндексная модель (доп. ограничения 1-го вида)	Трехиндексная модель (доп. ограничения 2-го вида)
3	2	2
3	1	2
3	2	3
3	3	3
3	2	2
3	3	4
4	59	2184
4	85	390
4	21	615
4	19	183
4	22	391
4	23	443
5	2043	21 649
5	414	3487
5	2678	14 911
5	37	13 325
5	68	19 858
5	50	18 432
6	2003	69 612
6	9306	308 456
6	21 669	181 926
6	466	46 370
6	9831	50 507
6	74	33 177

численными координатами пунктов (включая базу) и весами  $q_i$ . При  $n = 3, 4, 5, 6$  были сгенерированы по 6 примеров. Результаты вычислений приводятся в табл. 3.

Приведенные результаты показывают, что добавление в модель ограничений 1-го вида повышает или не ухудшает эффективность поиска решения во всех случаях, а добавление дополнительных ограничений 2-го вида приводит к увеличению или к неуменьшению числа итераций в 18 из 24 примеров.

Для проведения сравнительного анализа моделей для частного случая использовалась библиотека примеров для симметричных задач TSPLIB [13]. Частный случай реализован с помощью принятия весов всех грузов и вместимости равными 1 (в табл. 4: 2\_1 для трехиндексной линейной модели) и с помощью исключения ограничения на вместимость и включения равенств (27), (28) (в табл. 4: 1\_2 и 2\_2 для двух- и трехиндексной линейных моделей соответственно).

Координаты пунктов принимались согласно данным из библиотеки примеров:

- у примеров с четным числом точек последняя точка убирается;
- первая точка принимается за депо;
- первая половина оставшихся точек принимается за производителей ( $i$ ), вторая — потребителей ( $i + n$ );
- веса грузов и вместимость ТС принимались равными 1.

Наиболее эффективными моделями для частного случая являются модели, в которых были устранены ограничения на вместительность и введены дополнительные ограничения (27), (28):

1) трехиндексная линейная модель (до  $n = 25$ , пример rat99.tsp,  $n = 49$ , не был решен);

2) двухиндексная линейная модель (до  $n = 391$ , пример с  $n = 645$  не был решен).

## Заключение

Построены двух- и трехиндексная модели задачи о заборе и доставке грузов одним ТС, в том числе для частного случая, когда каждый груз должен быть сразу доставлен потребителю.

Проведена серия численных экспериментов с использованием оптимизационного пакета CPLEX для оценки эффективности рассмотренных моделей. Для общего случая задачи наиболее эффективной является трехиндексная линейная модель. Для частного случая задачи — двухиндексная линейная модель с соответствующими дополнительными ограничениями.

Представляется, что дальнейший прогресс связан как с совершенствованием моделей, в частности с введением дополнительных ограничений, так и с совершенствованием пакета CPLEX. Как отметила С. Archetti (Брешиа, Италия) на 3-м совещании Европейской рабочей группы VeRoLog (Осло, 2014 г) [14], CPLEX 11 (2007) работает почти в 30 000 раз быстрее, чем CPLEX 1 (1991).

## Список литературы

1. **Danzig G., Ramser J.** The Truck Dispatching Problem // Management Science. 1959. Vol. 6, Iss. 1. P. 80—91.
2. **Vehicle routing problem.** URL: <http://neo.lcc.uma.es/vrp/> (дата обращения: 11.11. 2017)
3. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part II: Transportations between customers and depot // Journal fur Betriebswirtschaft. 2008. Iss. 58. P. 21—51.
4. **Ruland K. S., Rodin E. Y.** The pickup and delivery problem: Faces and branch-and-cut algorithm // Computers and Mathematics with Applications. June 1997. Vol. 33, Iss. 12. P. 1—13.
5. **Renaud J., Boctor F. F., Ouenniche J.** A heuristic for the pickup and delivery traveling salesman problem // Computers and Operations Research. 2000. Vol. 27, Iss. 9. P. 905—916.
6. **Renaud J., Boctor F. F., Laporte G.** Perturbation heuristics for the pickup and delivery traveling salesman problem // Computers and Operations Research. 2002. Vol. 29, Iss. 9. P. 1129—1141.
7. **Dumas Y., Desrosiers J., Soumis F.** The pickup and delivery problem with time windows. European Journal of Operation Research. 1991. Vol. 54, Iss. 1. P. 7—22.
8. **Furtadoa M., Munaria P., Morabito R.** Pickup and delivery problem with time windows: a new compact two-index formulation / Technical Report. Production Engineering Department, Federal University of São Carlos, Brazil. July 2015. URL: [www.optimization-online.org/DB\\_HTML/2015/07/5022.html](http://www.optimization-online.org/DB_HTML/2015/07/5022.html) (дата обращения: 11.11.2017)
9. **Cordeau J.-F., Laporte G.** The dial-a-ride problem: models and algorithms // Annals of Operations Research. 2007. Vol. 153, N. 1. P. 29—46.

Таблица 4

Сравнение моделей для частного случая

$n$	Пример	Модель	Итерация	Время
7	ulysses16.tsp	1_2	24	0,3
7	ulysses16.tsp	2_1	557 972	170,0
7	ulysses16.tsp	2_2	93	0,7
14	bayg29.tsp	1_2	78	1,0
14	bayg29.tsp	2_2	831	6,0
25	eil51.tsp	1_2	136	1,0
25	eil51.tsp	2_2	2811	90,0
49	rat99.tsp	1_2	434	3,2
68	gr137.tsp	1_2	743	5,0
114	gr229.tsp	1_2	1882	8,7
287	rat575.tsp	1_2	7134	125
391	rat783.tsp	1_2	17 128	550

10. **Бронштейн Е. М., Гиндуллина Э. В., Гиндуллин Р. В.** Формализации задач погрузки и доставки // Вестник Южно-Уральского университета. Сер. Математика. Механика. Физика. 2017. Т. 9, № 1. С. 13–21.

11. **Miller C., Tucker A., Zemlin R.** Integer programming formulations and travelling salesman problems // Journal of the ACM. 1960. Vol. 7, N. 4. P. 326–329.

12. **Cordeau J.-F., Laporte G., Ropke S.** Recent models and algorithms for one-to-one pickup and delivery problems. In B. L. Golden, S. Raghavan, and E. A. Wasil (Eds.) // The Vehicle

Routing Problem, Latest Advances and Challenges. Springer, Boston. 2007. P. 327–357.

13. **MP-TESTDATA** — The TSPLIB Symmetric Traveling Salesman Problem Instances. URL: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/> (дата обращения: 11.11.2017)

14. **Archetti C.** Matheuristics for routing problems. 3-rd Workshop of European Working Group VeRoLog (Oslo, 2014 г). URL: [https://www.sintef.no/contentassets/cfb19ab9b7c74d03904c7746ee1d8e77/matheuristics\\_routing\\_verolog2014\\_new.pdf](https://www.sintef.no/contentassets/cfb19ab9b7c74d03904c7746ee1d8e77/matheuristics_routing_verolog2014_new.pdf) (дата обращения 12.01.2018)

**E. M. Bronstein**, Professor, e-mail: bro-efim@yandex.ru,  
Ufa State Aviation Technical University, Ufa,  
**R. V. Gindullin**, Assistant Professor, e-mail: gramiz@mail.ru  
Bashkir State University

## Solving Single Vehicle Routing Problem with Pick-Ups and Deliveries with CPLEX

*There is considered vehicle touring problem of forming a cycle of minimal length, that ensures delivery of cargo of various mass from producers to consumers with single vehicle. A cargo from a specific producer must be delivered to a specific consumer. Problem is called a single vehicle pickup-delivery problem (SPDP). Such problem can occur in carriage of passengers (e.g. taxi service). Particular case of the problem is considered, where each cargo must be delivered immediately to a consumer. Formalizations of both general and particular cases of SPDP are offered, and realized in CPLEX. Comparison of the efficiency of proposed formalizations for general case of the problem is conducted on randomly generated instances, and for particular case — on instances from TSPLIB.*

**Keywords:** vehicle routing, cycle, optimization, integer linear programming

### References

1. **Danzig G., Ramser J.** The Truck Dispatching Problem, Management Science, 1959, vol. 6, iss. 1, pp. 80–91.

2. **Vehicle** routing problem, available at: <http://neo.lcc.uma.es/vrp/> (date of accessed 11.11. 2017)

3. **Parragh S., Doerner K., Hartl R.** A survey on pickup and delivery problems. Part II: Transportations between customers and depot, *Journal fur Betriebswirtschaft*, 2008, iss. 58, pp. 21–51.

4. **Ruland K. S., Rodin E. Y.** The pickup and delivery problem: Faces and branch-and-cut algorithm, *Computers and Mathematics with Applications*, June 1997, vol. 33, iss. 12, pp. 1–13.

5. **Renaud J., Boctor F. F., Ouenniche J.** A heuristic for the pickup and delivery traveling salesman problem, *Computers and Operations Research*, 2000, vol. 27, iss. 9, pp. 905–916.

6. **Renaud J., Boctor F. F., Laporte G.** Perturbation heuristics for the pickup and delivery traveling salesman problem, *Computers and Operations Research*, 2002, vol. 29, iss. 9, pp. 1129–1141.

7. **Dumas Y., Desrosiers J., Soumis F.** The pickup and delivery problem with time windows, *European Journal of Operation Research*, 1991, vol. 54, iss. 1, pp. 7–22.

8. **Furtadoa M., Munaria P., Morabito R.** Pickup and delivery problem with time windows: a new compact two-index formulation, Technical Report. Production Engineering Department, Federal University of São Carlos, Brazil, July 2015, available

at: [www.optimization-online.org/DB\\_HTML/2015/07/5022.html](http://www.optimization-online.org/DB_HTML/2015/07/5022.html) (date of accessed 11.11.2017)

9. **Cordeau J.-F., Laporte G.** The dial-a-ride problem: models and algorithms, *Annals of Operations Research*, 2007, vol. 153, no. 1, pp. 29–46.

10. **Bronshtejn E. M., Gindullina Je. V., Gindullin R. V.** *Formalizacii zadach pogruzki i dostavki* (Formalizations of pickup and delivery problems), *Vestnik Juzhno-Ural'skogo universiteta. Serija: Matematika. Mehanika. Fizika* (News of South-Ural University. Series: Mathematics. Mechanics. Physics.), 2017, vol. 9, no.1, pp. 13–21 (in Russian).

11. **Miller C., Tucker A., Zemlin R.** Integer programming formulations and travelling salesman problems, *Journal of the ACM*, 1960, vol. 7, no. 4, pp. 326–329.

12. **Cordeau J.-F., Laporte G., Ropke S.** Recent models and algorithms for one-to-one pickup and delivery problems, In B. L. Golden, S. Raghavan, and E. A. Wasil (Eds.), *The Vehicle Routing Problem, Latest Advances and Challenges*, Springer, Boston, 2007, pp. 327–357.

13. **MP-TESTDATA** — The TSPLIB Symmetric Traveling Salesman Problem Instances, available at: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/> (accessed 11.11.2017)

14. **Archetti C.** Matheuristics for routing problems / 3-rd Workshop of European Working Group VeRoLog (Oslo, 2014), available at: [https://www.sintef.no/contentassets/cfb19ab9b7c74d03904c7746ee1d8e77/matheuristics\\_routing\\_verolog2014\\_new.pdf](https://www.sintef.no/contentassets/cfb19ab9b7c74d03904c7746ee1d8e77/matheuristics_routing_verolog2014_new.pdf) (date of accessed 12.01.2018)



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

### **Коммуникационно-ориентированная архитектура распределенных информационных систем на основе службы обработки защищенных сообщений**

*Рассматриваются принципы организации архитектуры распределенных информационных систем, в которой функции прикладных компонентов отделены от функций коммуникационной среды по организации сетевого взаимодействия и информационной защиты (коммуникационно-ориентированной архитектуры). Описывается подход к построению такой архитектуры на основе новой сетевой службы, ориентированной на обработку защищенных сетевых сообщений.*

**Ключевые слова:** распределенные системы, Web-технологии, интернет-технологии, информационное взаимодействие, информационная безопасность

#### **Введение**

Одна из самых важных тенденций развития сетевых технологий во все последние годы заключалась в постоянном повышении уровня и удобства прикладного программного интерфейса (API) и все большей "изоляции" прикладного пользователя от базисных сетевых понятий. Если еще лет 10—15 назад программирование сетевого взаимодействия на уровне TCP или UDP (т.е. в терминах системных вызовов connect, listen, select, accept и т.п. [1, 2]) применялось не столь уж редко, то сегодня оно уже практически не используется в прикладных разработках. Важную роль в продвижении этой тенденции сыграло создание сетевой архитектуры .NET и, в частности, технологии Web-сервисов. Благодаря поддержке в целом ряде современных систем программирования (MS Visual Studio, Borland C++ builder, Oracle developer suit и т.п.) и Web-серверов (IIS, Apache, Oracle HTTP server), а также удачным сетевым стандартам WSDL и SOAP этой технологии удалось быстро занять лидирующее место в создании сетевых приложений [3, 4]. Наверное, более всех сумели оценить ее достоинства разработчики распределенных информационных систем (РИС), которые как никогда прежде

смогли абстрагироваться от подробностей сетевого взаимодействия и сосредоточиться на содержательной стороне проектов.

Тем не менее разработчики РИС сталкиваются и сегодня с рядом проблем. Это связано с тем, что технология Web-сервисов не предлагает готовых решений в такой важной области, как защита и аутентификация данных в сети или же авторизация информационных запросов и разграничение прав доступа к сервисным компонентам (пожалуй, единственным готовым решением является использование защищенного протокола HTTPS в качестве "транспорта" вместе со всеми связанными с ним ограничениями: только одна электронная подпись в данных, невозможность сохранить информационный запрос или ответ вместе с удостоверяющими подписями и т.п.). Кроме того, технология Web-серверов вводит ряд собственных сетевых понятий и конструкций, таких как proxy-class, binding, contract endpoint и т.п., предназначенных для использования в клиентских и серверных компонентах РИС. Все это приводит к порочной практике "перемешивания" прикладных функций компонентов РИС с функциями коммуникационной среды и средств защиты, а в конечном итоге — к увеличению сроков разработки и снижению

уровня информационной безопасности [5, 6]. Важно подчеркнуть, что и сами Web-сервисы являются сетевыми программными модулями особого вида с собственными правилами разработки, отладки и конфигурирования.

В данной работе рассматривается подход к созданию достаточно простой сетевой архитектуры, свободной от отмеченных недостатков.NET. Другими словами, речь идет об архитектуре, ориентированной на защищенную обработку информационных запросов в РИС без использования специфических сетевых конструкций в исходном коде как клиентских, так и сервисных модулей. В целом, предлагаемый подход основан на осознании того, что сетевые средства взаимодействия следует рассматривать не просто как второстепенные по важности добавки к функциональным компонентам РИС, но как "несущую конструкцию", поддерживающую работу всей системы. Здесь можно провести аналогию с приборным шкафом (коммутационная среда), в который с помощью стандартного разъема вставляются аппаратные функциональные модули, не содержащие собственных коммуникационных средств. Сетевую архитектуру РИС, в которой функции прикладных компонент отделены от функций коммуникационной среды по организации сетевого взаимодействия и информационной защиты, будем называть коммуникационно-ориентированной архитектурой (КОА).

Предлагаемый метод организации КОА основан на двух основных понятиях (программных классах): понятии защищенного сетевого сообщения и понятии динамических библиотек сервисных функций, выполняющих обработку таких сообщений. Класс "Защищенное сообщение" отображает контейнер для хранения электронных документов (информационных запросов или ответов), снабженных одной или несколькими удостоверяющими электронными цифровыми подписями (ЭЦП). Отличительная особенность класса "Защищенное сообщение" заключается в том, что он объединяет в себе функции защиты данных с функциями их обработки. В отличие от технологии Web-сервисов, все сервисные функции имеют жесткую сигнатуру: они получают объект класса "Защищенное сообщение" в качестве параметра и возвращают объект того же класса в качестве результата. Разумеется, на электронный документ, содержащийся в этих объектах, не накладывается никаких ограничений (например, это может быть документ в

формате XML, содержащий любое число параметров или реквизитов различного типа). Существенным здесь является то, что организация работы с библиотекой функций, размещенной на удаленном сервере, внешне практически не отличается от организации работы с локальной библиотекой, размещенной на рабочей станции. От клиентской программы требуется лишь подключить библиотеку сервисных функций, подготовить исходящее сообщение, содержащее информационный запрос, осуществить вызов требуемой функции для обработки этого сообщения и получить результат выполнения запроса во входящем сообщении. Практически вся работа по установлению соединения с сервером, шифрованию данных в сети, формированию и проверке ЭЦП у исходящего и входящего сообщений и разграничению прав доступа к сервисным функциям выполняется вне кода клиентской программы и вне кода сервисных библиотек.

### **Короткий пример сетевого взаимодействия**

Описываемый подход к построению КОА основан на использовании сетевой службы PMS (Protected Message Service) [7] в качестве базисного средства организации сетевых взаимодействий. Хотя данное применение PMS потребовало доработки и расширения функциональности как клиентской библиотеки функций (PmsBase.dll), так и сервера PMS, свойства ее главного класса PmsMessage ("Защищенное сообщение") внешне практически не изменились. Доработанная служба PMS по-прежнему опирается непосредственно на TCP/IP и не использует никаких элементов Web-технологии (Web-серверы, HTTP, SOAP, WSDL и т.п.).

Рассмотрим простой пример сетевого взаимодействия в нотации языка C# (рис. 1). Предположим, что в каталоге обрабатывающих библиотек сервера PMS с интернет-именем mysgv.ru имеется динамическая библиотека MyLib.dll, в которой определена сервисная функция GetFileList, получающая в качестве параметра полное имя папки в файловой системе сервера и возвращающая список имен файлов, содержащихся в этой папке.

В первой строке примера определяется переменная Request класса PmsMessage. Эта переменная инициализируется значением строки, содержащей полное имя папки. Во второй строке к этой переменной добавляются две ЭЦП, сформированные с помощью функции-члена

```

PmsMessage Request= new PmsMessage("c:\\windows\\system");
Request.AddSignaturesByNames(new string [ ] {"Иванов", "Петров"});

//----- Вызов функции GetFileList из удаленной (серверной) библиотеки
PmsSrvLibraries RemLibs = new PmsSrvLibraries("mysrv.ru", "O=Мос*| O=Центр*");
if(RemLibs.ErrMsg != "") Console.WriteLine ("Ошибка: " + RemLibs.ErrMsg);
else
{
    PmsMessage Reply=Request.Process (RemLibs, "MyLib.GetFileList","T=Зав*");
    if(Reply != null)
        Console.WriteLine (Reply.GetString());
    else
        Console.WriteLine ("Ошибка: " + RemLibs.ErrMsg);
}
RemLibs.Close();

//----- Вызов той же функции из локальной библиотеки
PmsSrvLibraries LocLibs = new PmsSrvLibraries("c:\\MyLibs\\bin\\Debug");
if(LocLibs.ErrMsg != "") Console.WriteLine ("Ошибка: " + LocLibs.ErrMsg);
else
{
    PmsMessage Reply=Request.Process (LocLibs, "MyLib.GetFileList");
    if(Reply != null)
        Console.WriteLine (Reply.GetString());
    else
        Console.WriteLine ("Ошибка: " + LocLibs.ErrMsg);
}
LocLibs.Close();

```

**Рис. 1. Пример вызова сервисной функции**

(метода) `AddSignaturesByNames`, обеспечивающей поиск и выборку сертификатов подписантов непосредственно из хранилища пользователя по именам владельцев сертификатов ("Иванов" и "Петров"). Это означает, что с этими сертификатами (вернее, с содержащимися в них открытыми ключами) обязательно должны быть связаны парные им закрытые ключи, иначе формирование ЭЦП закончится аварийно (из примера намеренно удалены операторы обработки исключений).

Четвертая строка примера содержит определение и инициализацию переменной `RemLibs` класса `PmsSrvLibraries`, открывающей доступ к библиотекам сервисных функций. Используемый в примере конструктор класса `PmsSrvLibraries` содержит два строковых параметра, содержащих имя сервера PMS и контрольные значения реквизитов владельца, на основании которых будет определена валидность сертификата сервера.

Логика конструктора в данном случае включает установление сетевого соединения с сервером PMS и немедленный запрос сертификата сервера для последующего шифрования информационных запросов с помощью содержащегося в нем открытого ключа. Важно подчеркнуть, что полученный сертификат проходит не только обычную проверку на корректность (с использованием "доверенного" сертификата), но и проверку на соответствие контрольным значениям реквизитов владельца. В данном

примере клиент требует, чтобы название организации владельца сертификата начиналось со строк "Мос" или "Центр", за которыми может следовать все что угодно. В противном случае в открытый член `ErrMsg` будет занесено диагностическое сообщение о сфальсифицированном сервере. Как видно из примера, контрольные значения реквизитов задаются на основе ключевого формата стандарта X509 [6], но с возможностью использования символов-заместителей ("\*" — любая строка, "?" — любой символ) или регулярных выражений.

Если работа конструктора заканчивается успешно, то выполняется вызов сервисной функции с помощью метода `Process` класса `PmsMessage` с занесением результата обработки (ответа) в переменную `Reply` (строка 8). В качестве параметров метода задаются переменная `RemLibs`, открывающая доступ к удаленным библиотекам, полное имя сервисной функции в формате "имя\_библиотеки.имя\_функции" и, опционально, контрольные значения реквизитов подписантов в ответе сервера. В данном примере клиент требует, чтобы название должности подписанта начиналось со строки "Зав". Обработка запроса завершается выводом на консоль или полученного результата (списка имен файлов), извлеченного из переменной `Reply`, или сообщения об ошибке. Важно подчеркнуть, что дешифрование запроса и формирование ЭЦП для результата выполняются сервером PMS автоматически без участия сервисной функции на основании сертификатов, заданных в его конфигурации. Точно так же шифрование ответа перед отправкой в сеть сервер выполняет автоматически (с использованием сертификатов, извлеченных из ЭЦП, содержащихся в запросе). Дешифрование же ответа и проверка его подписей выполняются на рабочей станции в рамках метода `Process` и без участия программы клиента.

В нижней половине рис. 1 проиллюстрировано обращение к той же функции, но из локальной библиотеки, размещенной в каталоге "c:\\MyLibs\\bin\\Debug" рабочей станции. Внешне все выглядит почти точно так же, только вместо имени сервера в конструкторе `PmsSrvLibraries` указано имя каталога, а контрольные значения реквизитов сертификата отсутствуют, так как шифрование данных в таком случае не требуется. Однако логика обработки меняется радикально. Вместо установления соединения с сервером PMS конструктор выполняет поиск всех динамических

библиотек в указанном каталоге и поиск всех функций, принимающих объект класса PmsMessage в качестве параметра и возвращающих объект этого же класса в качестве результата, в каждой библиотеке. Информация обо всех найденных функциях сохраняется в специальной структуре данных класса PmsSrvLibraries, обеспечивающей быстрый поиск функции по полному имени и доступ к ней. Соответственно меняется и логика выполнения метода Process класса PmsMessage: вместо обращения к серверу PMS осуществляется вызов функции локальной библиотеки. Данный способ вызова предназначен для отладки сервисных функций и их взаимодействия с клиентом вне сетевой среды. Если на рабочей станции установлена среда разработки (например, MS Visual Studio) и имеются проекты клиентского модуля и библиотеки сервисных функций, то можно провести отладку функции с использованием всего инструментария отладочной среды, включая точки останова и пошаговое выполнение. Отлаженная библиотека может быть перенесена на сервер PMS (в специальный каталог серверных библиотек, заданный в конфигурации сервера) без каких-либо изменений.

Оператор RemLib.Close() закрывает сетевое соединение с сервером PMS, а оператор LocLib.Close() освобождает локальные ресурсы.

Определение сервисной функции для данного примера в нотации с# приведено на рис. 2. Здесь важно обратить внимание на сиг-

```
public class MyLib
{
    public PmsMessage GetFileList(PmsMessage Inp, string [ ] Conf, ref string Msg)
    {
        string sRes = "";
        try
        {
            string Dir = Inp.GetString();
            string [ ] sFiles = Directory.GetFiles(Dir);
            if (sFiles != null)
            {
                for (int i = 0; i < sFiles.Length; i++)
                    sRes += sFiles[i] + "\r\n";
            }
        }
        catch (Exception Ex)
        {
            Msg=Ex.Message;
            if(PmsBase.Cfg(Conf, "Debug")=="yes")
                (PmsBase.Log(PmsBase.Cfg(Conf, "Log"), Ex.Message);
            return null;
        }
        Msg = "OK";
        PmsMessage Res = new PmsMessage(sRes);
        return Res;
    }
    ...
}
```

Рис. 2. Пример определения сервисной функции

натуру функции: все сервисные функции независимо от назначения и логики обработки должны иметь одинаковую сигнатуру. Первым и главным параметром функции является объект класса PmsMessage, содержащий информационный запрос. Вторым параметром Conf содержит конфигурационные данные в форме массива строк вида "имя = значение". Через третий строковый параметр Msg функция передает диагностическое сообщение (которое становится доступным клиенту через член ErrMsg класса PmsSrvLibraries). Возвращаемое значение функции — объект класса PmsMessage, содержащий результат обработки. Сервисные функции, имеющие иную сигнатуру, сервер игнорирует.

При подключении каждой динамической библиотеки сервер PMS выполняет поиск ее конфигурационного файла, совпадающего по имени с файлом библиотеки, но имеющего расширение имени ".cfg" (своего рода аналог файла web.config для Web-сервисов). Этот файл может содержать определения строковых конфигурационных параметров в форме "имя = значение" как для библиотеки в целом, так и для каждой функции индивидуально. Конфигурационные данные сохраняются во внутренних структурах данных сервера и автоматически подаются в качестве значения параметра Conf на вход каждой функции при обращении к ней. Пример конфигурационного файла библиотеки приведен на рис. 3. Отметим, что индивидуальные параметры функции размещаются после заголовка вида "[имя\_функции]". Разрешается вводить параметры с любыми именами, но некоторые параметры имеют системное значение и интерпретируются сервером PMS. К последним относятся, например, параметры Timeout, определяющий тайм-аут времени выполнения функции в секундах; Access, обеспечивающий разграничение прав доступа к ней. Подчеркнем, что тайм-аут мож-

```
Log=c:\PmsLog\MyLib.log
...
[GetFileList]
Timeout=5
Access=T=Зав*,CN=Петров * | T=Зам*,CN=Сидоров *
Debug=yes

[GetFile]
Access=T=Директор*
...
```

Рис. 3. Пример конфигурационного файла библиотеки сервисных функций

но определять для каждой функции в отдельности (в отличие от Web-сервисов). Параметр Access задает набор контрольных значений реквизитов подписантов информационного запроса в уже знакомой нам нотации, на основе которого сервер PMS ограничивает доступ к той или иной функции. Например, доступ к функции GetFileList (см. рис. 3) открыт только тем запросам, среди подписантов которых имеется владелец сертификата с названием должности, начинающимся со строки "Зав", и фамилией "Петров" или же с названием должности, начинающимся со строки "Зам", и фамилией "Сидоров".

Как видно из рис. 2, функция GetFileList извлекает имя папки из параметра Inp и формирует список имен файлов, содержащихся в этой папке, в форме строки, используя статический метод GetFiles класса Directory. Полученный список помещается в новый объект класса PmsMessage и возвращается клиенту.

При возникновении исключительной ситуации (например, если указанная папка не существует) функция считывает значение конфигурационного параметра "Debug" из массива Conf с помощью статического метода PmsBase.Cfg, определенного в клиентской библиотеке PmsBase.dll. Если этот параметр имеет значение "yes", функция создает запись в файле журнала, определенном конфигурационным параметром "Log", с помощью статического метода PmsBase.Log.

## Протокол PMS

PMS в полной мере использует двоичную природу TCP/IP [7, 8]. Взаимодействие между клиентом и сервером PMS осуществляется по специальному, достаточно простому PMS-протоколу, ориентированному на передачу двоичных сетевых сообщений (PMS-сообщений) в обоих направлениях (никакие преобразования двоичных данных в текстовую форму типа base64 не применяются). Каждое такое сообщение в общем случае содержит два массива байтов: заголовок сообщения и тело сообщения (рис. 4). Первые 4 байта заголовка или тела сообщения содержат целое число — его длину. При передаче запроса от клиента к серверу в заголовок сетевого сообщения помещается строка, содержащая полное имя вызываемой функции, а в тело сообщения упаковывается структура PmsMessage в открытой или зашифрованной форме, содержащая информационный запрос.

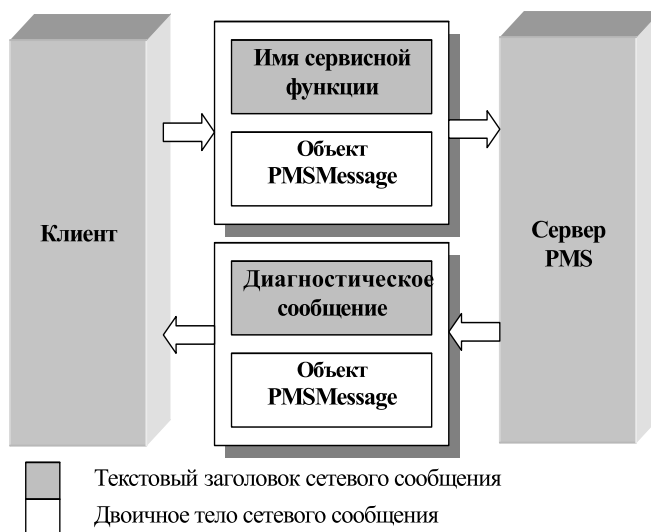


Рис. 4. Сетевые сообщения

Строку заголовка сервер использует для организации вызова соответствующей обрабатываемой функции. При передаче результата обработки от сервера к клиенту в заголовок сетевого сообщения помещается строка диагностического сообщения (значение параметра Msg, сформированное обрабатываемой функцией), а в тело сообщения упаковывается структура PmsMessage, содержащая ответ сервера в открытой или зашифрованной форме, предварительно подписанный собственным закрытым ключом сервера. Никакие двоично-текстовые преобразования (типа base64) не применяются. Полученное от сервера диагностическое сообщение автоматически присваивается члену ErrMsg объекта класса PmsSrvLibraries на стороне клиента (см. рис. 1).

## Классы PmsMessage и PmsSrvLibraries

Главная особенность класса PmsMessage заключается в том, что он содержит всего два элемента данных (поля):

- открытый (public) массив байтов Data, предназначенный для хранения передаваемых данных (т.е. "полезной нагрузки");
- массив электронных подписей для передаваемых данных, каждая из которых содержит не только собственно подпись ("свертку" данных, зашифрованную закрытым ключом подписанта), но и удостоверяющий сертификат, включающий парный открытый ключ и основные реквизиты подписанта (страна, город, организация, должность и т.п.).

Хотя класс PmsMessage включает довольно много функций-членов (для формирования,

проверки и удаления ЭЦП, для шифрования/дешифрования данных и для передачи их на обработку сервисным функциям), он содержит всего два конструктора: PmsMessage(byte [] Bytes) и PmsMessage(string Str), позволяющих инициализировать член Data данными из заданного массива данных или из символьной строки соответственно. Если пользователю нужно передавать объекты сложной структуры, то он должен самостоятельно запрограммировать средства "упаковки" таких объектов в массив байтов на отправляющей стороне и для "распаковки" на принимающей стороне (или же воспользоваться стандартными средствами байтовой сериализации).

Чтобы упростить задачу пользователя, в библиотеке PmsBase.dll определен класс PmsDocument, который можно рассматривать

```

class PmsDocument
{
    string Doc;           //текстовый документ
    Attachment [] AttArray; //массив приложений
    // конструктор («распаковка»)
    PmsDocument (PmsMessage Pmsg) {
        ...
    }
    // создание защищенного сообщения («упаковка»)
    PmsMessage GetPmsMessage() {
        ...
    }
}

class Attachment
{
    string Name;         //имя файла приложения
    string Comment;     //дополнительная информация
    byte [] Data;       //данные приложения
}
    
```

Рис. 5. Определение класса PmsDocument

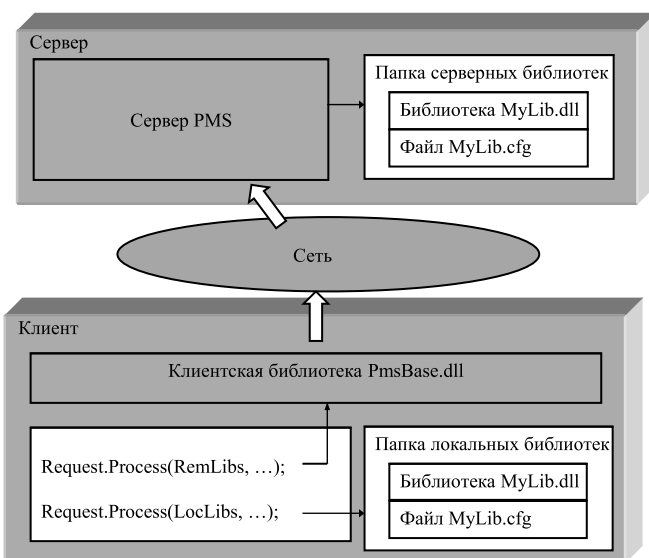


Рис. 6. Работа с удаленными и локальными библиотеками

как готовый пример программной "настройки" над классом PmsMessage. Этот класс предназначен для передачи информационных запросов и информационных ответов в РИС и содержит символьную строку для хранения текстового документа (например, в формате XML) и массив двоичных приложений к этому документу (графической информации, документов в формате PDF, MS-Word и т.п.). Определение класса на языке C# проиллюстрировано на рис. 5.

Алгоритмы "упаковки" объекта PmsDocument в массив байтов и "распаковки" в первоначальный вид (метод GetPmsMessage и конструктор PmsDocument) мы рассматривать не будем.

Логика работы с библиотеками сервисных функций, заложенная в класс PmsSrvLibraries, проиллюстрирована на рис. 6. Имена переменных Request, RemLibs и LocLibs в нижней части этого рисунка имеют то же смысл, что и на рис. 1.

Как видно из рис. 6, обращение к удаленной библиотеке сервисных функций через переменную RemLib выливается в передачу информационного запроса серверу PMS, который заранее (при запуске) просматривает свою папку серверных библиотек и собирает сведения обо всех найденных динамических библиотеках, сервисных функциях и их конфигурационных параметрах в своей внутренней структуре данных для обеспечения быстрого доступа к ним.

Обращение к локальной библиотеке сервисных функций через переменную LocLib выливается в прямой вызов функции из локальной динамической библиотеки без участия сервера PMS. Сбор сведений обо всех найденных динамических библиотеках, сервисных функциях и их конфигурационных параметрах выполняет конструктор PmsSrvLibraries при инициализации переменной LocLibs. Важно отметить, что в этом случае программа клиента и сервисная функция выполняются не только на одной машине, но и в контексте одного процесса, что очень удобно для отладки их взаимодействия.

Несколько слов о сетевых настройках. Все сетевые технологии, включая и Web-сервисы, допускают задание значений разнообразных сетевых настроек (размер буфера чтения, размер буфера записи, тайм-ауты ожидания и т.п.), но одновременно предлагают набор разумных значений по умолчанию, с тем чтобы пользователю почти никогда не пришлось иметь с ними дело. Как правило, эта цель достигается в отношении большинства настроек

ек, кроме одной: тайм-аута чтения данных из сети. Разработчикам РИС часто приходится "напрямую" управлять значением этого параметра, так как оно должно быть согласовано со временем обработки информационного запроса: если в клиентской программе тайм-аут чтения равен 120 с, а время обработки запроса на сервере может достигать 300 с, то пользователь рискует никогда не получить результата.

Конструктор PmsSrvLibraries также допускает задание сетевых настроек через дополнительные параметры, если их значения по умолчанию не удовлетворяют пользователя. Однако главная особенность PMS в части управления сетевыми настройками связана со значением тайм-аута чтения из сети в клиентской программе — это значение она принимает от сервера. Как уже отмечалось при обсуждении рис. 1, работа конструктора PmsSrvLibraries при подключении удаленных библиотек начинается с установления соединения с сервером и с запроса сертификата сервера. Предложенное решение основано на том, что ответ на этот запрос содержит не только сертификат, но и еще одну важную характеристику — максимальное значение тайм-аута выполнения сервисной функции, заданное в конфигурационных файлах серверных библиотек (конфигурационный параметр Timeout), увеличенное на 10 с. Если это значение больше клиентского тайм-аута чтения, то сетевое соединение с сервером закрывается, устанавливается новое значение тайм-аута чтения, равное полученному от сервера значению, после чего соединение с сервером немедленно восстанавливается, но уже с новой настройкой. Такое решение гарантирует, что клиент обязательно дожидется результата выполнения сервисной функции или же диагностического сообщения сервера о прерывании ее выполнения вследствие тайм-аута.

### Временные оценки

Для проведения лабораторных экспериментов и получения сравнительных оценок быстродействия была проведена реализация службы PMS в среде программирования Microsoft VisualStudio 2010 для платформы .Net Framework 4.0 на основе криптосистемы "КриптоПро CSP" версии 3.6, соответствующей требованиям действующих в России ГОСТов в области криптографической защиты информации.

Основная цель экспериментов с PMS заключалась в сравнении ее быстродействия с

быстродействием Web-сервисов в одинаковых условиях. Главное внимание уделялось вызовам сервисных функций с относительно малым (от нескольких миллисекунд до нескольких сотен миллисекунд) временем выполнения (при более длительной обработке разница между двумя технологиями практически нивелируется) с применением средств криптозащиты и без применения. В экспериментах с PMS использовали средства криптозащиты, интегрированные в клиентскую библиотеку и сервер PMS. В экспериментах с Web-сервисами средства криптосистемы "КриптоПро CSP" подключались непосредственно к программе клиента и программе Web-сервиса. При этом сервер PMS с модельными библиотечными функциями и Internet Information Server с модельными Web-сервисами были установлены на одном и том же четырехъядерном сервере приложений с тактовой частотой 2,4 ГГц в операционной среде Window 2003 Server. В качестве клиентской рабочей станции был использован одноядерный компьютер с тактовой частотой 2,8 ГГц.

На рис. 7—9 показаны характерные результаты экспериментов с очень быстрой сервисной функцией, выполняющей простое перекодирование полученного строчного сообщения в верхний регистр и возврат результата клиенту, при длине сообщения в 2, 50 и 100 Кбайт соответственно. На каждом рисунке приведены диаграммы времени выполнения операции на сервере с помощью Web-сервиса (черный столбик) и с помощью PMS (серый столбик) для четырех режимов: без применения криптозащиты; с применением ЭЦП; с применением ЭЦП и шифрования сообщений и, наконец, с применением ЭЦП, шифрования и проверки сертификатов на корректность. Проверку сертификатов выполняли путем построения "цепочки" сертификатов от проверяемого сертификата до "доверенного". В каждом режиме время выполнения вычисляли, как среднее значение для 100 последовательных вызовов сервисной функции.

Как видно из рис. 7—9, в данной серии экспериментов PMS не уступает технологии Web-сервисов в быстродействии и даже несколько превосходит ее. Причем в режиме "без криптозащиты" это превосходство является весьма значительным (что, по-видимому, объясняется временными затратами, связанными с применением протокола HTTP/SOAP). При подключении криптозащиты время обработки возрастает и различие становится менее существен-

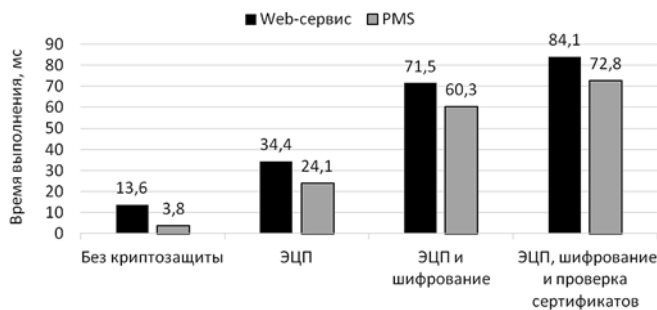


Рис. 7. Время обработки при длине сообщения 2 Кбайт

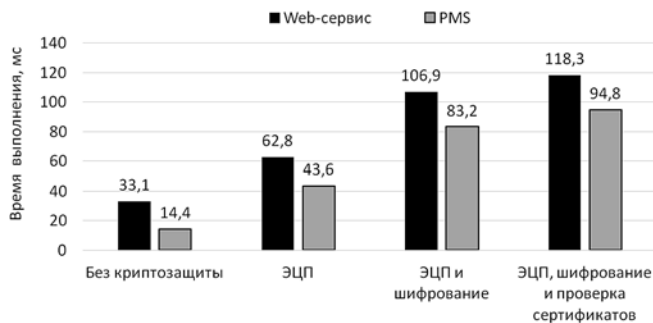


Рис. 8. Время обработки при длине сообщения 50 Кбайт

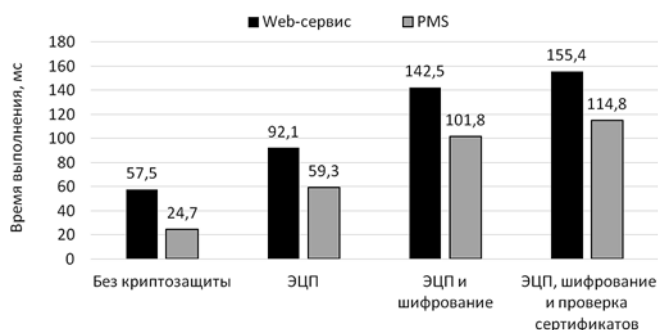


Рис. 9. Время обработки при длине сообщения 100 Кбайт

ным (характерно, что абсолютная разница во времени выполнения при этом остается относительно стабильной).

### Заключение

Фактически предложенный подход к построению КОА основан на отказе от гибкой модели организации взаимодействия, основанной на вызове методов удаленных объектов (например, через Web-сервисы), в пользу более жесткой модели, основанной на обмене защищенными сообщениями. Введение стандартной сигнатуры сервисных функций (получают объект класса "Защищенное сообщение"

в качестве главного параметра и возвращают объект этого же класса в качестве результата) не представляется слишком уж ограничительным и неудобным, когда речь идет об обработке информационных запросов в распределенных системах, так как оставляет возможность обмена электронными документами любой сложности. Вместе с тем эта унификация позволяет значительно продвинуться в направлении создания готовых решений таких важных для разработчика задач, как защита данных в сети, аутентификация и авторизация информационных запросов, разграничение прав доступа к сервисным функциям или же отладка клиентских и серверных компонентов вне сетевой среды. Основное же преимущество данного подхода заключается в том, что он позволяет сделать шаг в направлении главного требования КОА — разгрузке кода клиентских программ и библиотек сервисных функций от специфических программных конструкций, направленных на организацию безопасных взаимодействий в сети.

Если продолжить аналогию КОА с приборным шкафом, которая была использована во введении, то в описанном подходе классам PmsMessage и PmsSrvLibraries, по-видимому, выпадает роль "стандартного разъема" для подключения клиентских модулей в РИС (см. рис. 1), а жесткой сигнатуре сервисных функций — роль "стандартного разъема" для подключения серверных компонентов (см. рис. 2).

### Список литературы

1. Снейдер Й. Эффективное программирование TCP/IP. Библиотека программиста. СПб.: Символ-Плюс, 2002. 320 с.
2. Хант К. TCP/IP. Сетевое администрирование. — СПб.: Питер, 2007. 816 с.
3. Шапошников И. В. Web-сервисы Microsoft.NET. СПб: БХВ-Петербург, 2002. 336 с.
4. Мак-Дональд М., Шпуншта М. Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
5. Згоба А. И., Маркелов Д. В., Смирнов П. И. Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30—38.
6. Козлов А. Д., Орлов В. Л. Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. — М.: ИПУ РАН, 2017. — 156 с.
7. Асратян Р. Э. Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. 2016. № 11. С. 490—497.



## Communication-Oriented Architecture of Distributed Information Systems on the Base of Protected Message Service

*The principles of the organization of architecture of the distributed information systems in which functions of application-oriented components are separated from functions of the communication environment ("communication oriented" architecture) are considered. An approach to creation of such architecture on the basis of the new network service oriented on protected processing of the network messages (Protected Message Service — PMS) is describe. Distinctive feature of service is the close integration of authentication and data protection functions with functions of network information exchange. From the client point of view the service architecture is based on two main program classes: "Protected message" and "Remote Library Set". This classes offer necessary functionality not only for creating and protecting messages, but also for transferring them to remote server via established network connections for processing. Contrary to web-services, based on remote function call model, PMS-service is based on message processing model: all service functions receive object of "Protected message" class as a parameter and return another object of the same class as a result of processing. Main advantages of proposed service include high speed of processing (small overhead for data transmission) and a possibility of debugging both client, and server components of system out of the network environment. The experimental implementation of PMS in C# for Microsoft Framework 4.0 and the study of performance of new service (in comparison with web services in .NET architecture) were carried out and the area of its effective application is outlined. This area includes distributed systems in which requirements of high processing rate and information security are viewed more important, than flexibility in service functions specifications.*

**Keywords:** distributed systems, Web-technologies, Internet-technologies, network interactions, data security.

### References

1. Snader J. *Effektivnoe programirovanie TCP/IP* (Effective TCP/IP programming), SPb.: Simvol-Pljus, 2002. 320 p. (in Russian).
2. Hunt C. *TCP/IP. Setevoe administrirovanie* (TCP/IP Network administration), SPb.: Piter, 2007, 816 p. (in Russian).
3. Shaposhnikov I. V. *Web-servisy Microsoft.NET* (Web-services of Microsoft.NET), SPb: BHV-Peterburg (), 2002. 336 p. (in Russian).
4. MacDonald M., Szpuszta M. *Microsoft ASP.NET 3.5 s primerami na C# 2008 i Silverlight 2 dlja professionalov* (Pro Microsoft ASP.NET 3.5 in C# 2008 includes Silverlight 2), Moscow, Viljams, 2009. 1408 p. (in Russian).
5. Zgoba A. I., Markelov D. V., Smirnov P. I. Kiberbezopasnost: ugrozy, vyzovy, reshenija (Cybersafety: threats challenges, decisions), *Voprosy kiberbezopasnosti*, 2014, no. 5, pp. 30—38 (in Russian).
6. Kozlov A. D., Orlov V. L. *Metody i sredstva obespechenija informacionnoj bezopasnosti raspredelennyh korporativnyh sistem* (Methods and means of ensuring of information security in distributed enterprise systems), Moscow, IPU RAN, 2017, 156 p. (in Russian).
7. Asratian R. E. Internet-sluzhba zashhishhennoj obrabotki informacionnyh zaprosov v raspredelennyh sistemah (Internet service for protected information queries processing in distributed systems), *Programmnyaya inzheneria*, 2016, no. 11, pp. 490—497.

**Н. Т. Абдуллаев**<sup>1</sup>, канд. техн. наук, доц., e-mail: a.namik46@mail.ru,  
Азербайджанский технический университет, г. Баку,  
**М. М. Гасанкулиева**<sup>2</sup>, диссертант, e-mail: metahasanquliyeva@rambler.ru,  
**А. Д. Джабиева**<sup>2</sup>, канд. техн. наук, доц., e-mail: aynur.jabiyeva@outlook.com,  
Азербайджанский государственный университет нефти и промышленности, г. Баку

## Применение нейросетевой технологии при фликкер-шумовой спектроскопии электрокардиограммы

*В качестве метода нелинейной динамики, позволяющего извлекать заключенную в сигналах, продуцируемых организмом человека, информацию, рассматривается метод фликкер-шумовой спектроскопии. Новые возможности фликкер-шумовой спектроскопии в распознавании специфических особенностей биомедицинских сигналов обусловлены введением информационных параметров. Эти параметры, характеризующие составляющие исследуемых сигналов на разных частотных диапазонах, необходимы для расчета диагностических показателей. Автоматизацию процесса диагностирования функционального состояния сердечно-сосудистой системы предлагается реализовать с помощью искусственных нейронных сетей.*

**Ключевые слова:** фликкер-шум, спектроскопия, параметризация, диагностические показатели, хаотический сигнал, автокорреляционная функция, спектр мощности, нейронная сеть

### Введение

Динамика многих физиологических процессов, протекающих в организме человека, является хаотической и может быть описана с позиций теории нелинейных детерминированных систем. Хаотичность поведения сердечного ритма, как правило, связывается с деятельностью параметрической нервной системы [1—3]. В области исследования сердечно-сосудистой системы выделяют, в основном, методы анализа, математически применяемые к нестационарным сигналам, статистические свойства которых изменяются со временем. Часто они состоят из кратковременных высокочастотных компонентов, сопровождаемых длительными низкочастотными составляющими.

С учетом такого характера биоэлектрических потенциалов и, в частности, электрокардиографических сигналов наиболее приемлемым для их анализа может быть метод нелинейной динамики с вычислением количественных характеристик хаоса [4—6]. Анализ сигналов, имеющих такой динамический характер, осно-

вывается, как правило, на применении метода фликкер-шумовой спектроскопии, учитывающего в сложной динамической системе эффект перемежаемости (intermittency), который представляет собой смену относительно длительных участков ламинарного или регулярного во времени поведения характеристик неравновесной динамической системы, сингулярными участками хаотических всплесков и скачков [7—10].

Такая динамика характерна и для электрокардиограмм, имеющих ламинарную фазу, представленную слабыми изменениями (зубцы Р, Т, изолиния) переменной на относительно протяженных временных интервалах ( $T$ ), и резкие скачкообразные изменения ( $QRS$ -комплекс) на коротких временных интервалах ( $\tau$ ). Для извлечения информации из сложных сигналов в методе фликкер-шумовой спектроскопии используется автокорреляционная функция  $\psi(\tau)$ . Для классификации информации, заключенной в функции  $\psi(\tau)$ , как правило, используется не сама эта функция, а некоторые ее преобразования — спектр мощности  $s(f)$  и разностный момент  $\varphi^{(2)}(\tau)$  второго порядка. Информация,

извлекаемая из анализа зависимостей  $s(f)$  и  $\varphi^{(2)}(\tau)$ , построенных на основе временных рядов сигнала  $V(t_i)$ , позволяет получить ряд информативных параметров, которые имеют диагностическую ценность.

На основе вычислительного эксперимента были получены зависимости для нормального состояния сердечно-сосудистой системы и ряда "катастрофических" аритмий (желудочковая тахикардия, предсердная фибрилляция, предсердная аритмия). При этом использовались экспериментальные данные с общедоступного сайта [www.PhysioNet.org](http://www.PhysioNet.org).

### Определение информативных диагностических параметров

В результате анализа спектра мощности  $s(f)$  для сингулярной составляющей ЭКГ-сигнала получены информативные параметры:  $T_0$  — определяющее некоторое характерное время, в пределах которого реализуется взаимосвязь измеряемой динамической переменной  $V(t_i)$ ;  $n_0$  — безразмерный параметр, эффективно определяющий, каким образом эта взаимосвязь теряется по мере уменьшения частот до значений  $1/2\pi T_0$ ;  $s(0)$  — вклад в спектр мощности  $s(f)$ , определяемой наиболее высокочастотной сингулярной составляющей [11].

Параметризация регулярной составляющей ЭКГ-сигнала осуществляется с помощью выражения  $\varphi^{(2)}(\tau)$  с параметрами  $T_1$ ,  $\tau_1$  и  $H_1$  [12]. При этом параметр  $T_1$  определяет характерное время, на котором значения динамических переменных  $V(t_i)$  не коррелируют. Для получения достоверных значений дисперсии  $\sigma_1^2$  необходимо ее рассчитывать на временных интервалах, превышающих  $T_1$ . При этом параметр  $H_1$  (показатель Херста) показывает, по какому закону теряется взаимосвязь измеряемых в разные моменты времени значений  $V(t_i)$ .

Таким образом, при анализе сложного хаотического сигнала, каковым является ЭКГ-сигнал, рассматривается совокупность шести параметров, характеризующая корреляционные взаимосвязи в последовательностях нерегулярностей-"скачков" и нерегулярностей-"всплесков", свойственных данному сигналу.

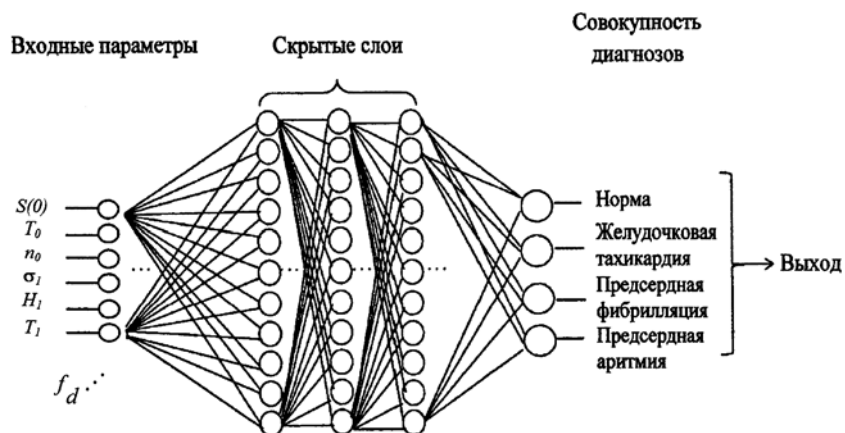


Рис. 1. Структурная схема предлагаемой искусственной нейронной сети

### Выбор искусственной нейронной сети и ее характеристики

Полученные значения совокупности параметров сингулярной и регулярной составляющих ЭКГ-сигналов могут быть использованы для дифференциальной диагностики функционального состояния сердечно-сосудистой системы с помощью искусственных нейронных сетей, где указанные параметры рассматриваются в качестве входных данных.

Для вычислительного эксперимента была выбрана персептронная трехслойная сеть с прямыми связями [13, 14] (рис. 1).

Для обучения нейронной сети был применен алгоритм обратного распространения ошибки (Back Propagation). Время обучения составило порядка 240 с, максимальная ошибка сети была порядка 0,05, степень обучения — порядка 0,01.

Для распознавания патологий сердечно-сосудистой системы может быть использован модульный вариант структуры построения нейросетевых блоков [15] (рис. 2).

Структура включает в себя несколько параллельно расположенных нейросетевых модулей, построенных на основе структуры многослойного персептрона. Преимуществом данной структуры является концентрация ресурсов каждого модуля на распознавании только одной патологии, что способствует уменьшению вероятности ошибки неверного заключения для всей системы в целом. Кроме того, расширяются функциональные возможности искусственной нейронной сети путем увеличения числа нейросетевых модулей для распознавания новых патологий без переобучения всей системы.

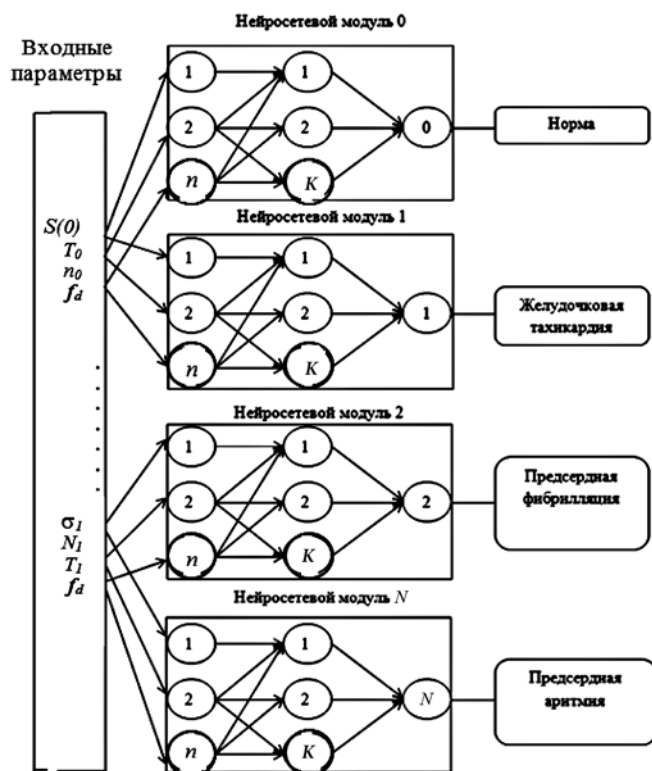


Рис. 2. Модульный вариант построения нейронной сети для распознавания патологий ( $n$  — число входных параметров;  $K$  — число нейронов промежуточного слоя;  $N$  — число анализируемых патологий)

Основным фактором, позволяющим реализовать выделение вклада нерегулярностей в анализируемые электрокардиографические сигналы, является вариация используемых частот дискретизации  $f_d$  реального сигнала [16]. Анализ зависимостей спектра мощности и разностного момента второго порядка, рассчитываемых на основе временных рядов с изменяющейся частотой дискретизации, позволяет для регулярной составляющей оценить меру "устойчивости", а для сингулярной составляющей — меру "вариабельности" ее информативных параметров. При этом параметр  $f_d$  может быть использован в качестве дополнительного входного параметра искусственной нейронной сети для распознавания состояний сердечно-сосудистой системы.

### Заключение

Представление электрокардиографических сигналов в виде последовательных нерегулярностей позволяет при анализе таких сигналов применять метод фликкер-шумовой спектроскопии. Хаотический сигнал, представленный временным рядом при фликкер-шумовой

спектроскопии, позволяет провести параметризацию этих сигналов и определить информативные диагностические показатели, характеризующие функциональное состояние сердечно-сосудистой системы. Совокупность информативных параметров, а также частота дискретизации сигнала, определяющая динамику изменения этих параметров, позволяет осуществлять классификацию сердечных заболеваний с помощью нейронной сети.

### Список литературы

1. Luther S., Eneart M., Luther G. et al. Nonlinear Dynamics and arrhythmia of the Heart. Max Plank Institute for Dynamics and self-Organization // Research Report. 2006. P. 84–86.
2. Баевский Р. М., Иванов Г. Г., Чирейкин Л. В. и др. Анализ вариабельности сердечного ритма при использовании различных электрокардиографических систем // Вестник аритмологии. 2001. № 24. С. 65–86.
3. Рангайян Р. М. Анализ биомедицинских сигналов. Практический подход. М.: Физматлит, 2007. 440 с.
4. Баевский Р. М., Кириллов О. И., Клецкин С. З. Математический анализ измерений сердечного ритма при стрессе. М.: Наука, 1984. 219 с.
5. Истомина Т. В., Ломтев Е. А., Сотникова О. А. Хаотическая динамика в нарушениях сердечного ритма // Медицинская техника. 1999. № 1. С. 34–35.
6. Coldberger A. L., Rigney D. R., West B. J. Chaos and fractals in human physiology // Scientific American. 1990. N. 2. P. 35–41.
7. Горбань И. И. Статистически неустойчивые процессы: связь с фликкер, неравновесными, фрактальными и цветными шумами // Изв. вузов "Радиоэлектроника". 2012. Т. 55, № 3. С. 3–16.
8. Parkhutik V., Rayon E., Ferrer C., Timashev S., Vstovsky G. Forecasting of electrical breakdown in porous silicon using flicker-noise spectroscopy // Physica Status Solidi (a). 2003. Vol. 197. P. 471–475.
9. Тимашев С. Ф. Фликкер-шумовая спектроскопия: информация в хаотических сигналах. М.: Физматлит, 2007. 248 с.
10. Тимашев С. Ф., Демин С. А., Панишев О. Ю., Поляков Ю. С., Каплан А. Я., Нефедьев Ю. А. Фликкер-шумовая спектроскопия как "прибор" для индивидуальной медицины будущего // "Ученые записки" Казанского Государственного Университета. Сер. "Физико-математические науки", 2012. Т. 154, № 4. С. 161–177.
11. Абдуллаев Н. Т., Дышин О. А., Гасанкулиева М. М. Фликкер-шумовая спектроскопия электрокардиографических сигналов // Медицинская техника. 2015. № 5. С. 12–15.
12. Абдуллаев Н. Т., Дышин О. А., Гасанкулиева М. М. Параметризация регулярной составляющей ЭКГ-сигнала для выявления критических состояний сердечно-сосудистой системы // Медицинская техника. 2016. № 3. С. 15–18.
13. Хайкин С. Нейронные сети: полный курс. М.: Издательский дом "Вильямс", 2006. 1104 с.
14. Ежов А. И., Четкин В. В. Нейронные сети в медицине // Открытые системы. 1997. № 4. С. 34–37.
15. Сушкова Л. Т., Исаков Р. В., Альмабрук М., Лукьянова Ю. А. Результаты исследования нейронных сетей в задачах распознавания патологических изменений электрической активности сердца // Биомедицинская радиоэлектроника. 2010. № 9. С. 9–13.
16. Абдуллаев Н. Т., Дышин О. А., Гасанкулиева М. М. Информационные параметры и флуктуационная динамика электрокардиограмм при фликкер-шумовой спектроскопии // Информационные технологии. 2016. № 1. С. 72–75.

## Application of Neuroet Network Technology for Flicker-Noise Spectroscopy of Electrocardiogram

The dynamics of many physiological processes occurring in the human body is chaotic and can be described from the positions of the theory of nonlinear deterministic systems. The randomness of the behavior of the heart rhythm, as a rule, is associated with the activity of the parametric nervous system. In the field of cardiovascular research, the analysis methods, mathematically applied to non-stationary signals, whose statistical properties change with time, are mainly isolated. Often they consist of short-time high-frequency components, accompanied by long low-frequency components. As a method of nonlinear dynamics, which makes it possible to extract the information contained in the signals produced by the human body, the method of flicker-noise spectroscopy is considered. New features of flicker-noise spectroscopy in the recognition specific features of biomedical signals are due to the introduction of information parameters. These parameters, which characterize the components of the signals under study at different frequency ranges, are necessary for the calculation of diagnostic indices. Automation of the process of diagnosing the functional state of the cardiovascular system is proposed to be realized with the help of artificial neural networks. Based on the computational experiment, dependencies were obtained for the normal state of the cardiovascular system and a number of "catastrophic" arrhythmias (ventricular tachycardia, atrial fibrillation, atrial arrhythmia). At the same time, experimental data were used from the public website [www.PhysioNet.org/](http://www.PhysioNet.org/) For the computational experiment, a perceptron three-layer network with direct links. To learn the neural network, the Back Propagation algorithm was applied. The training time was about 240 s, the maximum network error was of the order of 0.05, the degree of training was about 0.01. To recognize the pathologies of the cardiovascular system, a modular version of the neural network building structure can be used.

**Keywords:** flicker noise, spectroscopy, parametrization, diagnostic indices, chaotic signal, autocorrelation function, power spectrum, neural network

### References

1. Luther S., Enyeart M., Luther G. et al. Nonlinear Dynamics and arrhythmia of the Heart. Max Plank Institute for Dynamics and self-Organization, *Research Report*, 2006, p. 84–86.
2. Baevskiy R. M., Ivanov G. G., Chireykin L. V. i dr. *Analiz variablnosti serdechnogo ritma pri ispolzovanii razlichnykh elektrokardiograficheskikh system* (Analysis of heart rate variability when using various electrocardiographic systems), *Vestnik Aritmologii*, 2001, no. 24, pp. 65–86 (in Russian).
3. Rangayyan R. M. *Analiz biomeditsinskih signalov. Prakticheskiy podhod* (Analysis of biomedical signals. Practical approach), Moscow, Fizmatlit, 2007, 440 p. (in Russian).
4. Baevskiy R. M., Kirillov O. I., Kletskin S. Z. *Matematicheskii analiz izmereniy serdechnogo ritma pri stresse* (Mathematical analysis of heart rate measurements under stress) Moscow, Nauka, 1984, 219 p. (in Russian).
5. Istomina T. V., Lomtev E. A., Sotnikova O. A. *Haoticheskaya dinamika v narusheniyah serdechnogo ritma* (Chaotic dynamics in disorders of the heart rhythm), *Meditsinskaya Tehnika*, 1999, no.1, pp. 34–35 (in Russian).
6. Coldberger A. L., Rigney D. R., West B. J. Chaos and fractals in human physiology, *Scientific American*, 1990, no. 2, pp. 35–41.
7. Gorban I. I. *Statisticheski neustoychivyye protsessy: svyaz s flikker, neravnovesnyimi, fraktalnymi i tsvetnyimi shumami* (Statistically unstable processes: connection with flicker, nonequilibrium, fractal and color noises), *Izv. VUZov "Radioelektronika"*, 2012, vol. 55, no. 3, pp. 3–16 (in Russian).
8. Parkhutik V., Rayon E., Ferrer C., Timashev S., Vstovsky G. Forecasting of electrical breakdown in porous silicon using flicker-noise spectroscopy, *Physica Status Solidi (a)*, 2003, vol. 197, pp. 471–475.
9. Timashev S. F. *Flikker-shumovaya spektroskopiya: informatsiya v haoticheskikh signalah* (Flicker-noise spectroscopy: information in chaotic signals), Moscow, Fizmatlit, 2007, 248 p. (in Russian).
10. Timashev S. F., Demin S. A., Panishev O. Yu., Polyakov Yu. S., Kaplan A. Ya., Nefedev Yu. A. *Flikker-shumovaya spektroskopiya kak "pribor" dlya individualnoy meditsiny budushego* (Flicker-noise spectroscopy as an "instrument" for individual medicine of the future), *"Uchenye zapiski" Kazanskogo Gosudarstvennogo Universiteta. Ser. "Fiziko-matematicheskie nauki"*, 2012, vol. 154, no. 4, pp. 161–177 (in Russian).
11. Abdullaev N. T., Dyishin O. A., Gasankuliyeva M. M. *Flikker-shumovaya spektroskopiya elektrokardiograficheskikh signalov* (Flicker-noise spectroscopy of electrocardiographic signals), *Meditsinskaya Tehnika*, 2015, no. 5, pp. 12–15 (in Russian).
12. Abdullaev N. T., Dyishin O. A., Gasankuliyeva M. M. *Parametrizatsiya regulyarnoy sostavlyayushey EKG-signalu dlya vyyavleniya kriticheskikh sostoyaniy serdechno-sosudistoy sistemy* (Parametrization of the regular component of the ECG signal for the detection of critical states of the cardiovascular system), *Meditsinskaya Tehnika*, 2016, no. 3, pp. 15–18 (in Russian).
13. Haykin S. *Neyronnyie seti: polnyi kurs* (Neural networks: full course), Moscow, Publishing house "Vilyams", 2006, 1104 p.
14. Ezhov A. I., Chechetkin V. V. *Neyronnyie seti v meditsine* (Neural networks in medicine), *Otkrytyie sistemy*. 1997, no. 4, pp. 34–37 (in Russian).
15. Sushkova L. T., Isakov R. V., Almabruk M., Lukyanova Yu. A. *Rezultaty issledovaniya neyronnykh setey v zadachah raspoznvaniya patologicheskikh izmeneniy elektricheskoy aktivnosti serdtsa* (Results of the study of neural networks in problems of recognition of pathological changes in the electrical activity of the heart), *Biomeditsinskaya Radioelektronika*, 2010, no. 9, pp. 9–13 (in Russian).
16. Abdullaev N. T., Dyishin O. A., Gasankuliyeva M. M. *Informatsionnyie parametry i fluktuatsionnaya dinamika elektrokardiogramm pri flikker-shumovoy spektroskopii* (Information parameters and fluctuation dynamics of electrocardiograms in flicker-noise spectroscopy), *Informatsionnyie Tehnologii*, 2016, no. 1, pp. 72–75 (in Russian).

Д. В. Пантюхин, ст. преподаватель, e-mail: dpantiukhin@hse.ru,  
Е. Карелова, студент, e-mail: ekarelova@edu.hse.ru,  
Национальный исследовательский университет "Высшая школа экономики"

## Повышение качества классификации компьютерных атак сверточной нейронной сетью посредством балансировки обучающей выборки

*Рассматривается подход к балансированию обучающей выборки при решении задачи классификации компьютерных атак сверточной нейронной сетью. Проведено сравнение на одной и той же структуре нейронной сети качества классификации без и с использованием балансировки обучающей выборки. Проведен вычислительный эксперимент, показавший эффективность использования балансировки в случае, когда число данных разных классов в исходной выборке существенно различается.*

**Ключевые слова:** обнаружение компьютерных атак, сверточная нейронная сеть, балансировка обучающей выборки, базы данных компьютерных атак

### Введение

В настоящее время наблюдается бурный рост применений методов глубинного обучения в различных прикладных задачах классификации, кластеризации, прогнозирования и др. Несомненно, самые большие успехи глубинного обучения показаны в области обработки изображений, тем не менее эти методы находят свое применение и в других областях.

Повсеместное распространение компьютерных сетей делает их привлекательными для злоупотреблений, среди которых компьютерные атаки занимают существенное место. Обнаружение факта атаки на компьютерную сеть и определение ее типа являются актуальной и сложной задачей, что связано с ростом объемов передаваемой информации и повышением ценности этой информации. Эффективность классического подхода к обнаружению атак, связанного с составлением и обработкой так называемых сигнатур — признаков наличия атак, уменьшается в связи с тем, что вынужденно растет количество сигнатур, которые необходимо проверять для каждого сетевого соединения. Поэтому распространение получили методы, основанные на применении различных методов мягких вычислений. В настоящее время для обнаружения атак используются:

- сигнатурные методы [8—10];
- методы, основанные на спецификациях [11];
- методы анализа систем состояний [12—14];
- методы анализа графов сценариев атак [15];
- экспертные системы [16, 17];
- MARS — Multivariate Adaptive Regression Splines [18];

- SVM [19];
- иммунные сети [20, 21] и генетические алгоритмы;
- нейронные сети [22, 23].

Одними из наиболее успешных нейронных сетей стали сверточные нейронные сети, которые и будут использованы в данной работе.

Анализ публикаций по применению методов мягких вычислений для обнаружения компьютерных атак показывает, что подавляющее большинство исследователей используют в своей работе морально устаревшую базу данных KDD99 [6], которая была создана в 1998—1999 гг. и не в полной мере отражает современные атаки в компьютерных сетях.

Последние годы появляются новые базы данных компьютерных атак (см. разд. 1), однако число работ, в которых они используются, невелико. На наш взгляд, это связано с недостаточной информированностью исследователей о наличии новых баз данных, большинство из которых доступны для некоммерческого использования. Более подробно известные авторам базы данных компьютерных атак будут описаны в разд. 1, однако для всех них можно выделить особенность, связанную с природой компьютерной атаки. Во-первых, атака — это относительно редкое событие (за исключением DDOS-атак), которое теряется на фоне числа пакетов нормальных (не атакующих) сетевых соединений, во-вторых, число данных об атаках различного типа существенно отличается (например, в базе данных KDD99 число нормальных пакетов — 972780, число пакетов с атакой типа DDOS — 3883370, а с атакой типа U2R — всего 52).

Число записей разных типов атак в базе данных KDD99Cup

Файл БД	DoS	Probe	U2R	R2L	Normal
10 % KDD	391 458	4107	52	1126	97 277
CorrectedKDD	229 853	4166	70	16 347	60 593
Whole KDD	3 883 370	41 102	52	1126	972 780

Существенное различие в числе данных для атак различного типа приводит к тому, что классификатор должен учитывать эту информацию, иначе возможны ситуации, когда классификатор в среднем работает неплохо, но отдельные (редкие) типы атак не распознает вовсе. Один из возможных подходов к улучшению качества таких классификаторов заключается в размножении (дублировании) данных об атаках редких типов. В связи с тем, что современные сверточные нейронные сети основаны на "пакетном" обучении, когда нейронная сеть обучается на небольших блоках (пакетах, *batch*) данных, чередующихся во время обучения, имеет смысл поддерживать заданные пропорции числа данных различных классов внутри каждого пакета. Именно такой подход используется в нашей работе.

## 1. Базы данных компьютерных атак

Первой, самой популярной, а долгое время чуть ли не единственной свободно доступной базой данных компьютерных атак была база **KDD99Cup** [6], созданная в 1998—1999 гг. Это открытая, бесплатная для проведения экспериментальных исследований база данных, создана Калифорнийским университетом (*Information and Computer Science University of California*). Она содержит около 5 млн записей о сетевых соединениях. Каждая запись представляет собой образ сетевого соединения, включает 41 параметр сетевого трафика (например, такие как тип протокола, порт отправления и порт назначения и др.), среди которых содержится три типа признаков: символьные (например, тип протокола), логические (флаги) и числовые.

База содержит информацию о 22 типах атак, при этом атаки делятся на четыре основных класса: Denial of Service (DoS), Remote to User (R2L), User to Root (U2R) и Probing.

Атака *Denial of Service (DoS)* — отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию сервера и включает шесть типов: *back*, *land*, *neptune*, *pod*, *smurt*, *teardrop*.

Атака *Remote to User (R2L)* характеризуется получением доступа незарегистрированного пользователя к компьютеру со стороны удаленной машины, включает четыре типа: *buffer\_overflow*, *perl*, *loadmodule*, *rootkit*.

Атака *User to Root (U2R)* предполагает получение зарегистрированным пользователем привилегий локального суперпользователя

(администратора), включает восемь типов: *ftp\_write*, *guess\_passwd*, *imap*, *multihop*, *phf*, *spy*, *warezclient*, *warezmaster*.

Атака *Probing* заключается в сканировании портов в целях получения конфиденциальной информации и включает четыре типа: *ipsweep*, *nmap*, *portsweep*, *satant*.

Число записей, относящихся к разным типам атак, существенно различно и приведено в табл. 1. В этой базе доступно также несколько файлов в текстовом формате, представляющих собой модификации этой базы.

База была подвергнута критике [1], основными замечаниями к ней были:

- большое число дублирующихся бесполезных записей (около 80 %!);
- недостаток записей о редких типах атак;
- моральное устаревание — база собрана в 1998—1999 гг. и не содержит информации о современных атаках.

В 2009 г. (через 10 лет!) получила развитие также модификация этой базы под названием **NSL-KDD**, в которой были отброшены дублирующиеся записи, проведена работа по выравниванию числа данных о разных типах атак [1]. В результате база сократилась в 4 раза.

В 2012 г. была анонсирована база данных **UNB ISCX 2012** Intrusion Detection Evaluation Data Set [2], созданная в Канадском Институте кибербезопасности (The Canadian Institute for Cybersecurity (CIC)). Эта база, так же как и KDD99, содержит информацию о сетевых соединениях, но также предлагает записи всего трафика. Трафик получен в течении 7 дней с тестовой компьютерной сети (6 локальных подсетей, NAT сервер, основной и вторичный серверы, средства мониторинга трафика) при различных сценариях атак, включает 2450324 соединений (около 90 Гбайт трафика), из них 68792 — атаки.

В 2017 г. была представлена база **UNB ISCX 2017**, построенная по аналогичному принципу и содержащая новейшие сценарии атак. Также был представлен инструмент для анализа трафика [3].

В 2015 г. была представлена база **UNSW-NB15** [4, 5], созданная в Австралийском Цен-

тре Кибербезопасности (Australian Centre for Cyber Security). База содержит как записи о сетевых соединениях, так и сам трафик (около 100 Гбайт) с тестовой компьютерной сети (три сервера). В базе представлены девять типов атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms. Каждая запись содержит 47 полей информации о сетевых соединениях и два поля информации о типе атаки (или ее отсутствии). Общее число записей около 2 млн. Информация о сетевых соединениях разбита на четыре текстовых файла (формат csv), отдельно представлены файлы для обучения и тестирования классификаторов, содержащие соответственно 175 341 и 82 332 записи.

## 2. Сверточные нейронные сети для классификации компьютерных атак

### 2.1. Описание сверточных нейронных сетей

Сверточные нейронные сети состоят из нескольких вычислительных слоев следующего типа.

А. Сверточные слои — это слои, которые выполняют операцию свертки входных данных с неким "ядром" — набором параметров. Параметры таких ядер чаще всего могут изменяться в процессе обучения, однако встречаются и сверточные слои с неизменяющимися параметрами. К последним можно отнести так называемые слои "пулинга" (*pooling*) которые предназначены для резкого уменьшения числа настраиваемых параметров. В нашей работе, так как размерность входных данных не очень велика, слои пулинга использоваться не будут, и все свертки — с настраиваемыми параме-

трами. Операцию свертки можно применять к данным любой размерности, наибольшее распространение получили двумерные свертки, поскольку основная область применения — это обработка изображений. Однако для нашей задачи данные представляются в виде векторов, т.е. одномерных массивов, поэтому будем использовать одномерные свертки. Пусть  $x_i$  —  $i$ -й элемент входного вектора,  $i = 1 \dots N$ , где  $N$  — число входных параметров,  $w_k$  —  $k$ -й компонент ядра (которое также представлено в виде вектора),  $k = 1 \dots K$ , в ядро также входит отдельный коэффициент  $w_0$ , который называется "смещение", тогда выход одномерного сверточного слоя определяется как:

$$y_j = \sum_k (x_{i+k} * w_k) + w_0, j = 1 \dots N - K.$$

Схематично эта операция представлена на рис. 1, а.

В одном сверточном слое может обрабатываться сразу несколько разных ядер, тогда выход сверточного слоя является совокупностью выходов каждого из ядер.

Б. Полносвязные слои — это слои, состоящие из нейронов, каждый из которых умножает элементы своего входа на свой весовой коэффициент, складывает эти произведения и добавляет смещение. Входы во все нейроны одного слоя одинаковые. Пусть  $x_i$  —  $i$ -й элемент входного вектора,  $i = 1 \dots N$ , где  $N$  — число входных параметров;  $w_{i,j}$  —  $i$ -й весовой коэффициент  $j$ -го нейрона;  $w_{0,j}$  — смещение  $j$ -го нейрона, тогда выход полносвязного слоя определяется как

$$y_j = \sum_i (x_i * w_{i,j}) + w_{0,j}.$$

Схематично эта операция представлена на рис. 1, б.

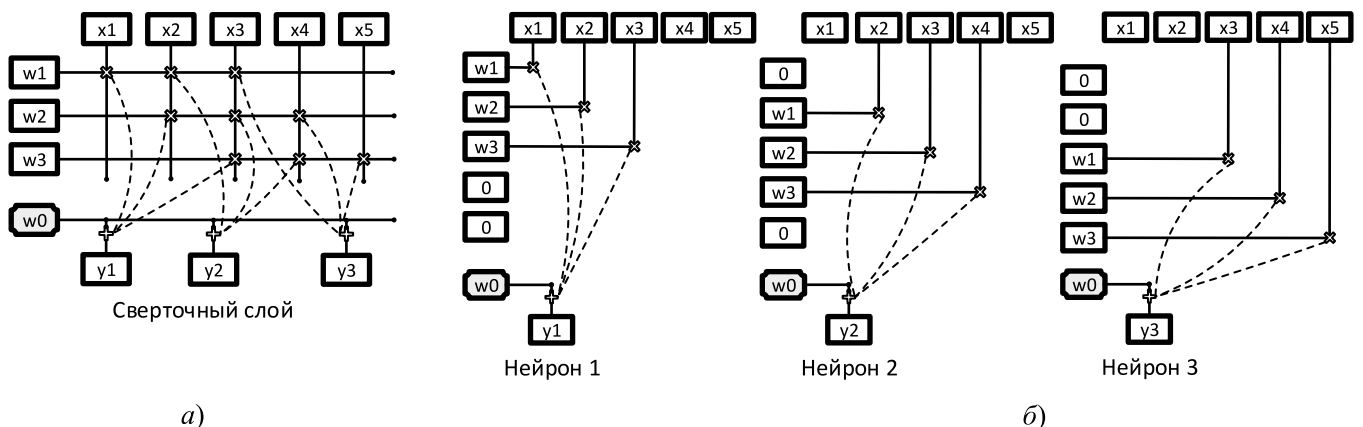


Рис. 1. Пример сверточного (а) и эквивалентного ему полносвязного (б) слоев нейронной сети для пяти входов и трех выходов: x — множитель; + — сумматор



Сравнив работу полносвязного и сверточного слоев, можно понять, что сверточный слой фактически является вариантом полносвязного слоя, в котором некоторые коэффициенты одинаковы для всех нейронов и многие из них равны 0 (что эквивалентно тому, что отсутствует связь от определенного входа, пример показан на рис. 1). За счет того что сверточные слои используют гораздо меньшее число параметров, чем полносвязные, но при этом их совокупность позволяет строить довольно сложные отношения вход-выход, они и получили широкое развитие.

Рассмотрим те типы функций активации, которые будем использовать в нашей работе.

1. *Сигмоидальная функция.* Эта функция описывается выражением: выход  $y = 1/(1 + e^{-x})$  для каждого входа  $x$  независимо.

2. *Softmax-функция.* Функция преобразует вектор входов  $x$  размерности  $N$  в вектор выходов  $y$  той же размерности, где каждый компонент выхода  $y_j$  полученного вектора представлен вещественным числом в интервале  $[0,1]$  и сумма компонент равна единице:

$$y_j = \exp(x_j) / \left[ \sum_i \exp(x_i) \right].$$

Представим требуемые выходы нейронной сети в виде бинарного вектора с числом компонент, равным числу классов, и в котором наличие единицы обозначает, к какому классу относится тот или иной пример. Тогда выходы этой функции активации можно трактовать как степень уверенности нейронной сети в том, что текущий пример данных имеет определенный класс. Обычно эта функция используется на последнем слое нейронной сети.

Вообще говоря, в сверточной нейронной сети могут присутствовать и другие типы слоев (например, слои с радиально-базисными нейронами, необучаемыми свертками и т.д.), но поскольку в нашей работе мы их не использовали, то не будем и описывать.

Сверточная сеть в целом состоит из набора чередующихся сверточных слоев и слоев активации, последние слои обычно полносвязные, также чередуются со слоями активации. В нашей работе использована сеть, схематичное изображение которой представлено на рис. 2.

## 2.2. Преобразование входных и выходных данных

Для того чтобы применить сверточную нейронную сеть для задачи классификации типов

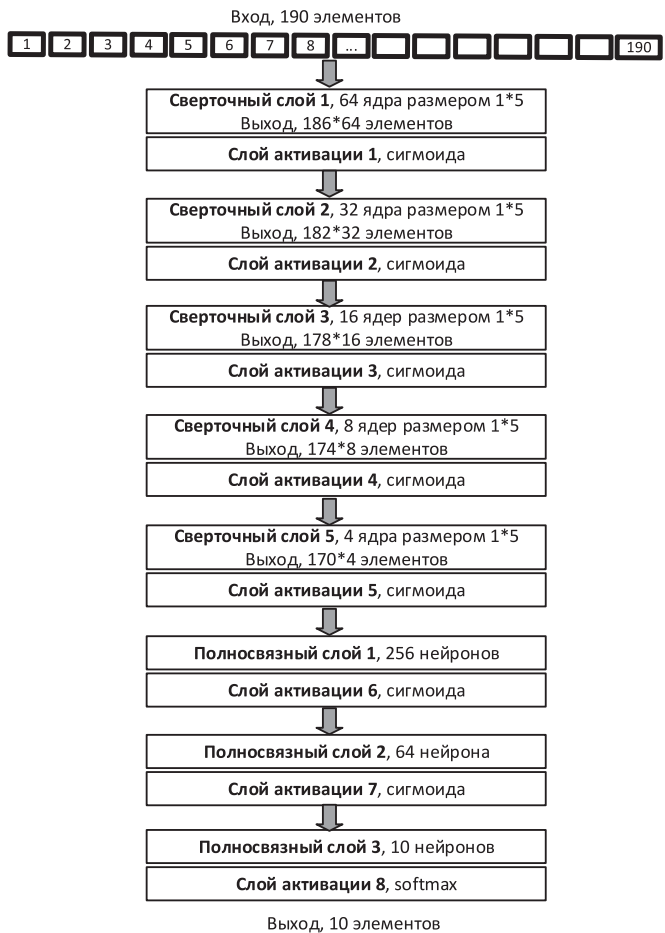


Рис. 2. Структура используемой нейронной сети

компьютерных атак, необходимо сначала представить данные из используемой базы данных в числовом виде, для чего нужно провести кодирование нечисловых данных, а также выполнить некоторую предобработку для числовых данных.

В используемой нами базе данных UNSW-NB15 каждый пример представляется 47 признаками, из них мы отбросим признаки, относящиеся только к той компьютерной сети, на которой собиралась база данных, а именно, следующие:

- srcip (source IP address) — IP-адрес источника,
- dstip (destination IP address) — IP-адрес получателя,
- sport (Source port number) — порт источника,
- dsport (destination port number) — порт получателя,
- stime (record start time) — время (абсолютное) начала записи,
- ltime (record last time) — время (абсолютное) последней записи,
- res\_bdy\_len — имеет отношение только к http сервису, не используется.

Из оставшихся признаков три являются категориальными:

- proto (Transaction protocol) — протокол, к которому относится сетевое соединение, может принимать одно из 129 значений;
- servis, указывает, какой сервис используется в сетевом соединении, может принимать значения: http, ftp, smtp, ssh, dns, ftp-data, irc и "-", если не используется;
- state, определяет состояние протокола, в зависимости от протокола может принимать значения: ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN и "-", если у протокола нет состояния.

Каждый из этих трех признаков будем кодировать с помощью бинарного вектора, число элементов в котором равно числу возможных значений признака. Все, кроме одного элемента такого вектора, нулевые, не нулевой элемент (равный единице) показывает, какое значение принял признак.

После кодирования число признаков стало 190 (37 — без изменений, 16 — для кодирования state, 8 — для servis и 129 — для proto).

Таблица 2

Масштабирование признаков

Название	Значение	Делитель
dbytes	Число байт переданных от получателя к источнику	100 000
dinpkt	Средняя задержка сетевых пакетов на получателе, мс	1000
djit	Джиттер (средние отклонения в задержке) сетевых пакетов на получателе, мс	1000
dload	Нагрузка получателя, бит/с	100 000
dloss	Число пакетов отброшенных или повторно переданных для получателя	10
dmean	Средний размер пакета от получателя, байт	10
dpkts	Число пакетов получатель-источник	100
dtepb	Длина базы пакета TCP протокола получателя	10 000 000
sinpkt	Средняя задержка сетевых пакетов на источнике, мс	1000
sjit	Джиттер (средние отклонения в задержке) сетевых пакетов на источнике, мс	10 000
sload	Нагрузка источника, бит/с	10 000 000
sloss	Число пакетов отброшенных или повторно переданных для источника	10
smean	Средний размер пакета от источника, байт	10
spkts	Число пакетов источник-получатель	100
stcpb	Длина базы пакета TCP протокола источника	10 000 000

Для оставшихся признаков проведены следующие преобразования:

- все бинарные признаки оставлены без изменений,
- часть признаков, приведенных в табл. 2, отмасштабированы путем деления на коэффициент,
- остальные признаки оставлены без изменений.

### 2.3 Формирование обучающей выборки с балансировкой классов и без

Выходы нейронной сети представляют собой 10-мерные векторы, по числу классов: девять атак (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms) и класс нормальных пакетов.

Нейронная сеть обучается алгоритмом RMS-проп библиотеки keras [7], одним из вариантов метода градиентного спуска, при этом обучение ведется на небольших батчах (batch, пакет) в двух вариантах, схематично показанных на рис. 3:

— батчи формируются из всех имеющихся примеров для обучения случайно, число примеров разных классов в батче не контролируется;

— число примеров разных классов в батче задано (в нашей работе одинаково), внутри класса примеры выбираются случайно.

Далее мы сравним результаты классификации этих двух вариантов.

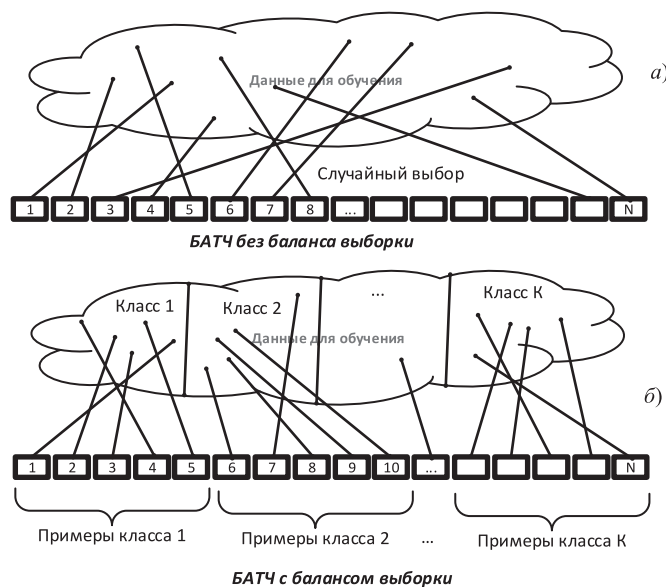


Рис. 3. Пояснения к способам формирования обучающей выборки:

а — формирование батча без баланса; б — формирование батча с балансом классов в выборке

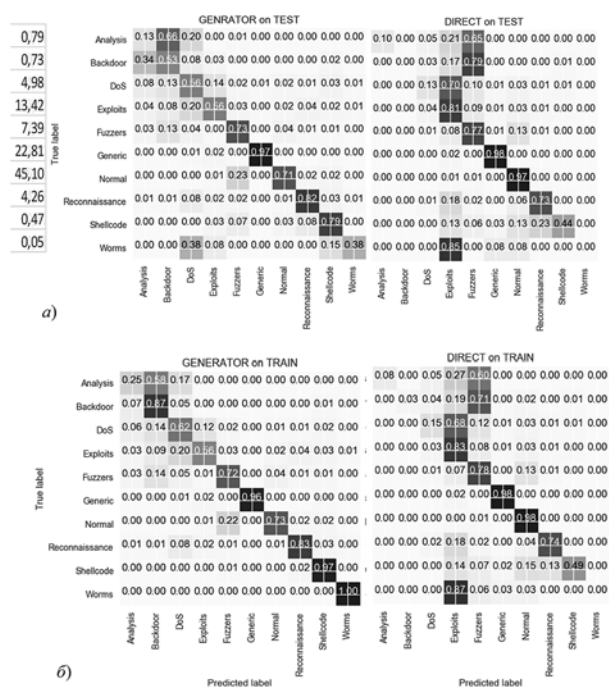
### 3. Вычислительные эксперименты

Для создания, обучения, отображения результатов использованы следующие программные средства:

- язык программирования: Python 3.5 в составе пакета Anaconda 4.2.0;
  - среда разработки: Microsoft Visual Studio 2017, v15.5.2;
  - библиотека обучения нейронных сетей: keras 2.0.2 на основе tensorflow (GPU версия) 1.1.0;
  - вспомогательные библиотеки: pandas 0.20.1 — для работы с текстовыми файлами базы данных; numpy 1.13.1 — для расчетов, scikit-learn 0.18.1 — для преобразования данных (кодирование), matplotlib 2.0.2 — для отображения графиков;
- и следующие аппаратные средства:
- ЭВМ с процессором IntelCore i7, 3,4 ГГц, 32 Гб оперативной памяти, под управлением операционной системы Microsoft Windows 7 Pro,
  - графический ускоритель (GPU): Nvidia Quadro K600, 1 Гб.

В рамках экспериментов использованы такие параметры нейронной сети:

- размер батча `batch_size` — 256;



**Рис. 4. Результаты экспериментов — матрицы потерь:**  
*a* — матрицы потерь на тестовых данных; слева (GENERATOR on TEST) — при использовании балансировки классов, справа (DIRECT on TEST) — без балансировки; *б* — матрицы потерь на обучающих данных; слева (GENERATOR on TRAIN) — при использовании балансировки классов, справа (DIRECT on TRAIN) — без балансировки

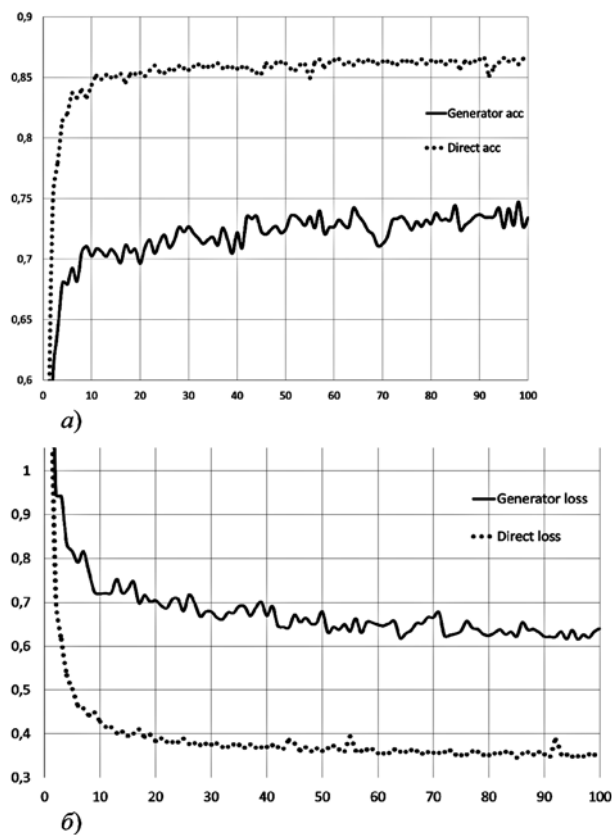
— максимальное число эпох (итераций) обучения — 200;

— шаг в алгоритме обучения  $l_r = 0,001$ .

Прочие параметры по умолчанию библиотеки keras [7].

Обучение велось в двух вариантах, с балансировкой обучающей выборки и без. Обучение повторялось несколько раз, приведены средние результаты.

На рис. 4 и 5 представлены результаты экспериментов. На рис. 4 приведены матрицы потерь для всех классов. Матрица потерь показывает, сколько данных было отнесено к тому или иному классу, по вертикальной оси приведены истинные классы данных, по горизонтальной оси — классы, предсказанные нейронной сетью на тестовых данных (отличных от обучающих). Правильная классификация отображена диагональными элементами матрицы потерь, неправильная классификация — недиагональными элементами. Результаты представлены в нормализованном виде, т.е. в долях от числа данных конкретного класса. Перед метками



**Рис. 5. Результаты экспериментов (первые 100 итераций):**  
*a* — точность (ассигасу), для способа без балансировки (пунктирная линия, Direct acc) и с балансировкой (сплошная линия, Generator acc); *б* — функция потерь (loss) для способа без балансировки (пунктирная линия, Direct loss) и с балансировкой (сплошная линия, Generator loss)

классов указано число данных каждого класса в процентах.

На рис. 5 приведены графики изменения точности (accuracy) классификации и функции потерь (losses). Точность вычисляется как отношение числа правильно распознанных примеров к общему числу примеров в тестовой выборке, функция потерь — это функция, которая оптимизируется в процессе обучения нейронной сети, в нашем случае это функция кросс-энтропии (categorical\_crossentropy, см [7]).

Анализируя результаты, можно сделать вывод, что применение балансировки помогает улучшить качество классификации данных с малым числом примеров. Отсутствие балансировки ведет к тому, что классы с малым числом примеров могут быть полностью проигнорированы нейронной сетью. Из рис. 5 видно: несмотря на то, что без использования балансировки была достигнута более высокая точность (accuracy) и меньшее значение функции потерь (loss), качество распознавания классов с малым числом примеров довольно мало, классы Worm, Backdoor не распознались вовсе.

Отметим также, что ценой за лучшее качество распознавания классов с малым числом примеров стало снижение качества распознавания классов с большим числом примеров (Exploits, Normal).

### Заключение

В работе показан подход к балансировке классов в задачах классификации с разным числом данных в классах, такой подход позволяет повысить качество классификации. В дальнейшем планируется повысить качество классификации за счет усложнения нейронной сети (увеличения числа слоев, подбора более оптимальных параметров слоев), добавления самообучающихся слоев.

*Авторы выражают благодарность всем коллегам за обсуждение результатов работы и особенно Научному фонду НИУ ВШЭ, поскольку публикация подготовлена в ходе проведения исследования (№ 16-01-0012 "Исследование применения сверточной нейронной сети для классификации типов компьютерных атак, используя базу данных компьютерных атак") в рамках Программы "Научный фонд Национального исследовательского университета "Высшая школа экономики" (НИУ ВШЭ)" в 2016—2018 гг. и в рамках государственной поддержки ведущих университетов Российской Федерации "5-100".*

1. **Tavallae M., Bagheri E., Lu W., Ghorbani A.** A detailed analysis of the KDD CUP 99 data set // Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009 IEEE Symposium on. IEEE, 2009. pp. 1—6.
2. **Shiravi A., Shiravi H., Tavallae M., Ghorbani A. A.** Toward developing a systematic approach to generate benchmark datasets for intrusion detection // Computers & Security. 2012. V. 31. No. 3. P. 357—374.
3. **Sharafaldin I., Lashkari A. H., Ghorbani A. A.** Toward generating a new intrusion detection dataset and intrusion traffic characterization // 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal. January 2018. P. 108—116.
4. **Nour M., Slay J.** UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) // Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015. P. 1—6.
5. **Nour M., Slay J.** The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set // Information Security Journal: A Global Perspective. 2016. Vol. 25, N. 1—3. P. 18—31.
6. **KDD Cup 1999 Data** [Электронный ресурс]. URL: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (дата обращения: 12.01.2018).
7. **Keras**: The Python Deep Learning Library [Электронный ресурс]. URL: <https://keras.io/> (дата обращения: 12.01.2018).
8. **Kumar S., Spafford E. H.** A pattern matching model for misuse intrusion detection // Proceedings of the 17th National Computer Security Conference. Baltimore MD, USA. 1995. P. 11—21.
9. **Kumar S., Spafford E. H.** An application of pattern matching in intrusion detection. The COAST Project, Department of Computer Sciences, Purdue University, West Lafayette, IN, USA, Technical Report CSD-TR-94-013. 1994. 57 p.
10. **Teng H. S., Chen K., Lu S. C.** Adaptive real-time anomaly detection using inductively generated sequential patterns // Proceedings 1990 IEEE Computer Society Symposium on Research in Security and Privacy, 1990. IEEE, 1990. P. 278—284.
11. **Ko C. C. W.** Execution monitoring of security-critical programs in a distributed system: a specification-based approach. PhD thesis, University of California, Davis, 1996. 118 p.
12. **Eckmann S. T., Vigna G., Kemmerer R. A.** STATL: An attack language for state-based intrusion detection // Journal of Computer Security. 2002. Vol. 10, No. 1—2. P. 71—103.
13. **Ho Y.** Partial order state transition analysis for an intrusion detection system // Master's thesis, University of Idaho, USA, 1997.
14. **Ilgun K., Kemmerer R. A., Porras P. A.** State transition analysis: a rule-based intrusion detection approach // IEEE Transactions on Software Engineering. 1995. V. 21, No. 3. P. 181—199.
15. **Sheyner O. M.** Scenario Graphs and Attack Graphs. PhD thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh PA, USA, 2004.
16. **Sebring M., Shellhouse E., Hanna M., Whitehurst R.** Expert systems in intrusion detection: a case study // Proc. 11th National Computer Security Conference, Baltimore, Maryland, Oct. 1988. 1988. P. 74—81.
17. **Whitehurst R. A.** Expert systems in intrusion detection: a case study. Report of Computer Science Laboratory, SRI International, Menlo Park, CA, USA, 1987.
18. **Smaha S. E.** Haystack: An intrusion detection system // Aerospace Computer Security Applications Conference, 1988, Washington, D. C., USA. IEEE 1988. P. 37—44.
19. **Mukkamala S., Sung A. H., Abraham A.** Intrusion detection using an ensemble of intelligent paradigms // Journal of Network and Computer Applications. 2005. Vol. 28, No. 2. P. 167—182.

20. **Hofmeyr S. A., Forrest S.** An immunological model of distributed detection and its application to computer security. PhD thesis, University of New Mexico. 1999.
21. **Zielinski M., Venter L.** Applying mobile agents in an immune-system-based intrusion detection system: reviewed article // *South African Computer Journal*. 2005. No. 34. P. 76–83.
22. **Debar H., Becker M., Siboni D.** A neural network component for an intrusion detection system // *Proceedings of 1992*

IEEE Computer Society Symposium on Research in Security and Privacy, 1992. IEEE 1992. P. 240–250.

23. **Jalili R.** et al. Detection of distributed denial of service attacks using statistical pre-processor and unsupervised neural networks // *International Conference on Information Security Practice and Experience*. Springer, Berlin, Heidelberg, 2005. P. 192–203.

**D. V. Pantiukhin**, Senior Lecturer, e-mail: dpantiukhin@hse.ru,

**E. Karelova**, Student, e-mail: ekarelova@edu.hse.ru,

National Research University Higher School of Economics, Moscow

## Quality Improvement of Intrusion Classification through Convolution Neural Network by Training on Balanced Samples

*This research investigates the effects of training sample balancing while solving intrusion classification task with convolution neural network. Using two convolutional neural networks with similar architecture, we conduct comparative analysis of classification task solution quality with and without training sample balancing. Experiments illustrate the efficiency of using training sample balancing in case of significant differences in the amount of samples in different classes.*

**Keywords:** intrusion detection, convolution neural network, training sample balancing, intrusion detection datasets

### References

- Tavallae M., Bagheri E., Lu W., Ghorbani A.** A detailed analysis of the KDD CUP 99 data set, *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009 IEEE Symposium on*. IEEE, 2009. pp. 1–6.
- Shiravi A., Shiravi H., Tavallae M., Ghorbani A. A.** Toward developing a systematic approach to generate benchmark datasets for intrusion detection, *Computers & Security*. 2012. vol. 31, no. 3, pp. 357–374.
- Sharafaldin I., Lashkari A. H., Ghorbani A. A.** Toward generating a new intrusion detection dataset and intrusion traffic characterization, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018, pp. 108–116.
- Nour M., Slay J.** UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), Military Communications and Information Systems Conference (MilCIS), 2015, IEEE, 2015, pp. 1–6.
- Nour M., Slay J.** The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, *Information Security Journal: A Global Perspective*, 2016, vol. 25, no. 1–3, pp. 18–31.
- KDD Cup 1999 Data**, available at: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (date of access: 12.01.2018).
- Keras**: The Python Deep Learning Library, available at: <https://keras.io/> (date of access: 12.01.2018).
- Kumar S., Spafford E. H.** A pattern matching model for misuse intrusion detection, *Proceedings of the 17th National Computer Security Conference*. Baltimore MD, USA, 1995, pp. 11–21.
- Kumar S., Spafford E. H.** An application of pattern matching in intrusion detection, The COAST Project, Department of Computer Sciences, Purdue University, West Lafayette, IN, USA, Technical Report CSD-TR-94-013, 1994, 57 p.
- Teng H. S., Chen K., Lu S. C.** Adaptive real-time anomaly detection using inductively generated sequential patterns, *Proceedings 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, 1990, IEEE, 1990, pp. 278–284.
- Ko C. C. W.** Execution monitoring of security-critical programs in a distributed system: a specification-based approach, PhD thesis, University of California, Davis, 1996, 118 p.
- Eckmann S. T., Vigna G., Kemmerer R. A.** STATL: An attack language for state-based intrusion detection, *Journal of Computer Security*, 2002, vol. 10, no 1–2, pp. 71–103.
- Ho Y.** Partial order state transition analysis for an intrusion detection system, Master's thesis, University of Idaho, USA, 1997.
- Ilgun K., Kemmerer R. A., Porras P. A.** State transition analysis: a rule-based intrusion detection approach, *IEEE Transactions on Software Engineering*, 1995, vol. 21, no. 3, pp. 181–199.
- Sheyner O. M.** Scenario Graphs and Attack Graphs, PhD thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh PA, USA, 2004.
- Sebring M., Shellhouse E., Hanna M., Whitehurst R.** Expert systems in intrusion detection: a case study, *Proc. 11th National Computer Security Conference*, Baltimore, Maryland, Oct. 1988, 1988, pp. 74–81.
- Whitehurst R. A.** *Expert systems in intrusion detection: a case study*, Report of Computer Science Laboratory, SRI International, Menlo Park, CA, USA, 1987.
- Smaha S. E.** Haystack: An intrusion detection system, *Aerospace Computer Security Applications Conference*, 1988, Washington, D. C., USA. IEEE 1988, pp. 37–44.
- Mukkamala S., Sung A. H., Abraham A.** Intrusion detection using an ensemble of intelligent paradigms, *Journal of Network and Computer Applications*. 2005, vol. 28, no. 2, pp. 167–182.
- Hofmeyr S. A., Forrest S.** An immunological model of distributed detection and its application to computer security, PhD thesis, University of New Mexico. 1999.
- Zielinski M., Venter L.** Applying mobile agents in an immune-system-based intrusion detection system: reviewed article, *South African Computer Journal*, 2005, no. 34, pp. 76–83.
- Debar H., Becker M., Siboni D.** A neural network component for an intrusion detection system, *Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 1992. IEEE 1992, pp. 240–250.
- Jalili R., Imani-Mehr F., Amini M., Shahriari H. R.** Detection of distributed denial of service attacks using statistical pre-processor and unsupervised neural networks, *International Conference on Information Security Practice and Experience*, Springer, Berlin, Heidelberg, 2005, pp. 192–203.

**С. Л. Макаров**, канд. техн. наук, доц., доц. Департамента программной инженерии  
Факультета компьютерных наук, e-mail: smakarov@hse.ru,  
Национальный исследовательский университет "Высшая школа экономики"

## Информационные технологии поиска аналогов исследования по каталогам диссертаций

*Рассматриваются некоторые информационные технологии поиска аналогов исследования, проекта или идеи по каталогам диссертаций и предлагается система автоматического поиска аналогов. Обоснована актуальность разработки системы, описаны методика и методы, лежащие в основе работы системы, и приведены результаты вспомогательных исследований структуры каталогов диссертаций и результаты работы системы. Приводится архитектура системы, структура базы данных, с которой работает система, описание алгоритма работы системы. Рассматривается пользовательский интерфейс и результаты работы системы с точки зрения пользователя и администратора системы. Делаются выводы о преимуществах и недостатках системы и особенностях ее работы и работы с ней, также описывается практическая значимость системы.*

**Ключевые слова:** информационные технологии, автоматический поиск, автоматизация, программная инженерия, система поиска, анализ проектов, аннотация исследования, каталог диссертаций, пользовательский интерфейс

### Введение

Для поиска аналогов исследования, проекта или идеи в настоящее время используются поисковые системы различных интернет-ресурсов. При использовании обычных информационно-поисковых систем (Яндекс, Google, Mail, Bing и т. д.) существует проблема поискового шума: вероятность найти реальное научное исследование невелика; скорее всего, будут найдены сайты магазинов, форумы, ссылки на бесчисленные блоги и социальные сети и прочие подобные ресурсы. Логично использовать для рассматриваемого типа поиска другие информационные технологии — специализированные ресурсы, исключающие или сводящие к минимуму эффект поискового шума. Такими ресурсами являются каталоги библиотек, в частности, каталоги диссертаций, каталоги научных статей, базы данных, индексирующие научные труды (Scopus, Web of Science, eLibrary и прочие). Однако значимость и вес научного исследования в диссертациях гораздо выше, чем в статьях или трудах конференции; кроме того, многие индексирующие базы данных закрыты извне — воспользоваться ими можно, лишь находясь в институте или университете. Поэтому в данной статье рассматривается поиск именно по каталогам диссертаций.

Система СТАРТLite (Старт Лайт) представляет собой веб-приложение и предоставляет пользователю возможность поиска существующих аналогов предлагаемого проекта или исследования, краткое описание (аннотацию) которого пользователь вводит в систему, по каталогам диссертаций. Среди аналогов могут быть найдены исследования, степень совпадения которых с описанием проекта пользователя является существенной. Таким образом, пользователь может оценить оригинальность своего исследования или проекта, получить первое представление о существующих конкурентах и сделать предварительный вывод о степени новизны своего проекта или идеи для исследования.

Здесь может возникнуть вопрос: почему не воспользоваться существующими каталогами диссертаций напрямую, например — поиском на сайте Российской государственной библиотеки? Во-первых, с помощью подобных каталогов нельзя ввести в поиск текст, можно лишь небольшую фразу или словосочетание, содержащее ключевые термины, интересующие пользователя. Во-вторых, поиск в подобных каталогах является достаточно сложной задачей — нужно разбираться, что именно искать, в каком именно каталоге, по какому полю — по теме, или по заглавию исследова-

ния, или по содержанию, если оно доступно; с учетом или без учета морфологии; изучать информационно-поисковый язык расширенных запросов и т. д. В-третьих, не всегда доступен полный текст или хотя бы оглавление закрытого ресурса, что довольно странно, учитывая, например, постоянную доступность подобной ознакомительной информации, например, для книг на сайте amazon.com. И, наконец, последнее (по порядку, но не по важности): для того чтобы найти что-то хотя бы по двум терминам, например "искусственный интеллект", нужно просмотреть десятки тысяч документов, даже если ограничивать поиск соответствующими фильтрами по техническим наукам, по языку документа и пр. Человеку такая задача не под силу. Предлагаемая система может осуществлять такой поиск автоматически, значительно быстрее человека. Безусловно, у аналогов системы есть и достоинства: возможность искать документы по различным полям: автору, ISBN, ISSN, регистрационному номеру, шифру специальности, системному номеру. Однако и здесь есть проблема — никто заранее не знает значений большинства этих полей.

#### Методика и методы, реализованные в системе

Для поиска аналогов проекта пользователя система использует набор из методики и методов автоматической обработки текстовых документов, таких как методика автоматического разбора текста с учетом морфологии [1, 2], метод автоматического построения векторной модели документа (простой вектор [3]), метод автоматического построения поисковых запросов с учетом архитектуры каталогов диссертаций, метод вычисления процента совпадения найденных документов с входными данными (косинусная мера [4]). Для поиска аналогов используются два каталога диссертаций: каталог [5] и ресурс Российской государственной библиотеки [6]. Оба ресурса являются открытыми для любого пользователя. Чтобы исключить поисковый шум, поиск ведется только по базе данных диссертаций.

Для морфологического разбора текстовых документов на русском языке в системе используется свободно распространяемый бинарный словарь и модуль на языке php, управляющий этим словарем. Словарь адаптирован к использованию в онлайн-режиме группой разработчиков открытого проекта phpMorphy [7], существующего уже достаточно давно. Именно этот модуль реализует методику автоматического разбора текста с учетом морфологии. С его помощью из текста выбираются наиболее

значимые и характерные термины — имена существительные и прилагательные. Исследование разработанной системы показало, что результаты поиска с учетом только имен существительных значительно проигрывают учету как существительных, так и характеризующих их прилагательных. Одной из проблем рассматриваемого модуля является неверное срабатывание на определенные слова — неправильное определение частей речи.

Метод автоматического построения векторной модели документа состоит в учете значимых терминов документа по отношению ко всему документу и базируется на частотном анализе этих терминов. При этом существительные и прилагательные смешиваются в один вектор документа, упорядоченный по убыванию частоты терминов:  $V = \{t_1, t_2, \dots, t_n\}$ , где  $t_1$  — наиболее часто встречаемый термин документа. Проблемой при таком подходе является тот факт, что в заголовках исследований и в их тексте нередко употребляются термины общего назначения, очень мало говорящие о сути исследования, например: система, подход, вид, тип, аннотация и тому подобные. Поэтому необходимо использовать список стоп-слов, куда нужно добавлять подобные малоинформативные термины. На данный момент система использует список стоп-слов, содержащий 82 элемента.

Метод автоматического построения поисковых запросов с учетом архитектуры каталогов диссертаций основывается на исследовании, проведенном для того, чтобы понять, как обращаться с ресурсами [5, 6]. Метод заключается в последовательном формировании поисковых запросов по каждому термину из вектора входных данных в виде адресной строки, которая посылается в систему ресурса [6], обрабатывается, и результат выполнения запроса затем разбирается на очередные 20 источников — названий и ссылок на диссертации, из которых берутся только названия. По этим названиям автоматически формируются ссылки к ресурсу [5] с помощью замены пробелов между словами на знак "+". Адресная строка ресурса [6] содержит очередной термин для поиска и вспомогательные параметры, например: `http://sigla.rsl.ru/results.jsp?f=1016&t=3&v0=интеллект&f=1003&t=1&v1=&f=4&t=2&v2=&x=42&y=11&f=21&t=3&v3=&f=1016&t=3&v4=&f=1016&t=3&v5=&bf=4&b=&d=0&ys=&ye=&lng=&ft=&mt=&doi=&dt=&vol=&pt=&iss=&ps=&pe=&tr=Cyr-Com mon&tro=&cc=a1&i=1&v=tagged&s=2&ss=1003&st=0&i18n=ru&psz=20&bs=20&ce=0w*M&d ebug=false&c=1&c=c3&c=b3&c=b2&c=b4&c=c4&c=b5&c=b6`. Как видно, ресурс [6] исполь-

зует GET-запросы, поэтому в адресной строке браузера располагаются все необходимые параметры, которые используются системой при поиске. В частности, значение параметра  $v0$  — это термин или термины, разделенные знаком +, которые пользователь ввел в строку поиска, а параметр  $i$  равен не номеру страницы, которую пользователь в данный момент просматривает на сайте в результатах поиска, а номеру первого источника, который располагается на странице. Если учесть, что ресурс выводит 20 источников на странице результатов поиска по умолчанию, то  $i$  будет равен 1 для первой страницы, 21 — для второй и так далее. Благодаря такой структуре ресурса разрабатываемая система может листать страницы поиска и менять поисковые термины, что и используется в рассматриваемом методе. Остальные параметры адресной строки являются служебными и не представляют интереса. Ресурс [5] служит для того, чтобы, скомпоновав название очередной диссертации, полученное с помощью работы с ресурсом [6], автоматически сформировать запрос и ссылку, которая предоставляется пользователю рядом с названием документа и перенаправляет пользователя на краткое содержание документа с оглавлением и возможностью прочитать автореферат, а в некоторых случаях и весь документ.

Метод вычисления процента совпадения найденных документов с входными данными основан на сравнении векторов очередного документа ( $d_2$ ) и входной информации, переданной пользователем системе ( $d_1$ ), с помощью определения расстояния между ними с использованием известной формулы косинусов или определения меры схожести двух документов [4]:

$$\text{sim}(d_1, d_2) = \frac{\sum_{j=1}^{N_t} d_{1j} d_{2j}}{\sqrt{\sum_{j=1}^{N_t} d_{1j}^2 \sum_{j=1}^{N_t} d_{2j}^2}},$$

где  $d_1 = (d_{11}, d_{12}, \dots, d_{1N_t})$  и  $d_2 = (d_{21}, d_{22}, \dots, d_{2N_t})$  — векторы документов  $d_{1j}$  и  $d_{2j}$ ,  $j = \overline{1, N_t}$  — значения  $j$ -го термина в документах  $d_1$  и  $d_2$ ;  $N_t$  — общее число разных терминов в обоих документах. Значение, полученное в результате такого вычисления, умножается на 100, так как выражено в процентах.

### Архитектура системы

Система состоит из нескольких модулей (рис. 1, см. третью сторону обложки): модуля для отправки входных данных (index.php), модуля поиска, запускаемого через планировщик

и осуществляющего обработку очереди заявок, поиск аналогов, запись результатов поиска и сравнение аналогов с введенными данными в базу данных (novizna.php), модулей отображения результата для пользователя (result.php) и администратора системы (resultadmin.php), модуля подробного отображения результатов для пользователя и администратора (resultkarta.php), модуля интерфейса изменения настроек (settings.php), модуля загрузки настроек и загрузки списка стоп-слов (setsloaded.php) и модулей, связанных со словарем phpMorphy, расположенных в папке tm2 (бинарный морфологический словарь русского языка и система управления этим словарем). В качестве вспомогательных файлов системой используются: директория js\_nova (файлы шаблона, необязательная директория), директория Temp (папка с файлом docounter.txt отображения хода процесса и параметрами перезапуска системы по расписанию), текстовый файл stoplist.txt, содержащий список стоп-слов, и текстовый файл settings.txt (файл настроек системы).

Кроме файловой части система работает с базой данных, состоящей из трех таблиц (рис. 2, см. третью сторону обложки): morph\_links (id "заказов", названия, ссылки, процент совпадения входных данных и документа по ссылке и термины документа по ссылке с их частотами), morph\_orders (id и входные данные "заказов" на поиск от пользователя системы, статус выполнения "заказов"), morph\_results (id "заказов", названия, ссылки, термины аналога с частотами и статус выполнения "заказов"). Термины, хранящиеся в таблицах, доступны для просмотра только администратором системы, причем совпадающие термины входных данных и аналога в интерфейсе выделяются жирным шрифтом. Таблицы базы данных не связаны друг с другом для простоты, однако понятно, что первичный ключ у таблиц для определенного заказа один и тот же, и все операции по выборке, обновлению или добавлению записей в таблицы осуществляются именно по нему.

Для обеспечения работы системы необходимо установить задание для планировщика операционной системы по запуску файла novizna.php через определенный промежуток времени. Рекомендуемый промежуток — 15...20 мин. Для linux-подобных систем это выглядит как создание нового задания для cron.

### Интерфейс и алгоритм работы системы

Алгоритм работы системы следующий (см. также [8]). Пользователь заходит на сайт и вво-



дит данные с помощью модуля для отправки входных данных. В качестве входных данных пользователь указывает свой адрес электронной почты для оповещения о готовности результатов, название своего проекта, ключевые слова проекта и аннотацию проекта, которая представляет собой краткое описание предлагаемого решения. Для запуска поиска необходимо заполнить все поля без исключения и затем нажать кнопку "Начать поиск" (рис. 3).

Если все поля заполнены правильно (есть проверка на корректность входных данных), пользователь видит сообщение, показанное на рис. 4, а система принимает введенные данные для обработки, записывает их в базу данных, пополняя таблицу заказов очередной записью, и начинает работу. При этом введенные данные сохраняются под определенным номером, который указывается в этом сообщении, пользователю предоставляется ссылка на результат работы системы, а его данные становятся в очередь. Под работой системы имеется в виду запуск модуля поиска системы с помощью планировщика, который находит необработанный заказ со входными данными в базе данных, строит для него векторную модель с помощью морфологического модуля, выделяет все термины и ищет каждый термин через каталог диссертаций, при этом для каждого найденного документа строит векторную модель, сравнивает ее с моделью входных данных и выдает процент совпадения, который записывает в базу данных. При превышении времени работы системы (600 с) она записывает все параметры ("то место, на котором остановилась") в файл счетчика и "засыпает", пока ее не "разбудит" планировщик. При "пробуждении" модуль поиска считывает все параметры и продолжает работу до тех пор, пока все термины в векторе входных данных не будут перебраны. По завершении работы система высылает письмо со ссылкой, по которой можно посмотреть результат поиска, на адрес, оставленный пользователем. При этом адрес электронной почты, введенный пользователем, не виден ни ему, ни администратору системы и известен только самой системе и ее разработчику, который может зайти в базу данных напрямую через панель управления хостинга.

Форма выходных данных или результатов работы системы, на которую пользователю была дана ссылка (рис. 4), содержит информацию обо всех заказах, содержащихся в системе (рис. 5).

Номер заявки соответствует номеру, который был присвоен введенным данным и сообщен пользователю (см. рис. 4), название проекта — введенному названию, и последнее поле — статус заявки — информирует пользо-

вателя о том, была ли его информация обработана (done) или обработка еще не закончена (undone). Пользователю предоставляется возможность нажать на название своего проекта и увидеть предварительные или окончательные результаты работы системы, которые отображаются в карте проекта (рис. 6).

Карта проекта отображает исходные данные: номер заявки, название проекта, ключевые слова проекта, аннотацию проекта и статус заявки, а также упорядоченные по убыванию степени совпадения аналоги проекта, найденные в каталоге диссертаций, и их число (рис. 6). Для каждого аналога выводится его название, автоматически сформированная ссылка на аналог на ресурсе [5], содержание аналога (в данной версии системы оно совпадает с названием, вообще же в идеале имеется в виду полный текст документа) и степень совпадения аналога с исходными данными, выраженная в процентах. В соответствии с настройками системы, которые может редактировать администратор системы, регулируется число выводимых пользователю результатов (в данном случае равное 333) из общего числа найденных аналогов на момент просмотра карты (в данном случае равное 920).

Для администратора системы форма результатов работы системы содержит дополнительную ссылку "Удалить" (рис. 7), при нажатии на которую из всех таблиц базы данных удаляется информация о соответствующем результате работы системы, и ссылку "Настройки системы", с помощью которой можно поменять параметры

Название проекта: Аппаратно-программный комплекс для оптимизированного управления специализированными координатными столами лазерной и плазменной резки

Ключевые слова проекта: астатизм второго порядка, числовое программное управление, минимизация динамической ошибки, системы автоматического управления

Аннотация проекта: Предложен проект для развития российского производства нового класса систем управления, главным отличием которых является то, что входной сигнал — астатизированный (точнее, частично (ограничено по времени) дегатеризированный), позволяет использовать следующие входные значения для минимизации взвешенной суммы ошибок. В связи с этим становится возможным использование системы для резки тонкого металла в лазерах с

Укажите Ваш email: somemail@mail.ru

Начать поиск

Рис. 3. Интерфейс заполнения входных данных

Спасибо за использование системы! Вашей заявке присвоен номер 1. Через некоторое время можно будет посмотреть результаты анализа [по этому адресу](#), а также Вам придет письмо на почту с этим адресом по завершении работы системы.

Рис. 4. Сообщение об успешном принятии введенных данных

Список проектов: [Назад](#)

Номер заявки	Название проекта	Статус заявки
1	<a href="#">Аппаратно-программный комплекс для оптимизированного управления специализированными координатными столами лазерной и плазменной резки</a>	undone
30	<a href="#">первый проект</a>	undone

Рис. 5. Форма результатов работы системы для пользователя

Данные проекта:

[Назад](#)

Номер заявки	Название проекта	Ключевые слова проекта	Аннотация проекта	Статус заявки
30	первый проект	методы интеллектуальной обработки информации	автоматизация анализа проектов с применением методов интеллектуальной обработки информации	undone

Список аналогов (всего найдено 920, из них отображено 333):

Название	Ссылка	Содержание	Степень совпадения
"Психическая болезнь" как интеллектуальный проект	<a href="#">ссылка</a>	"Психическая болезнь" как интеллектуальный проект	3.59%
Разработка и исследование системы обработки технологической информации гидрогенерирующих предприятий	<a href="#">ссылка</a>	Разработка и исследование системы обработки технологической информации гидрогенерирующих предприятий	3.59%
Автоматизация лингвистической обработки словарей научно-технической информации	<a href="#">ссылка</a>	Автоматизация лингвистической обработки словарей научно-технической информации	2.94%
Формирование учетно-контрольной информации для природоохранных проектов	<a href="#">ссылка</a>	Формирование учетно-контрольной информации для природоохранных проектов	2.94%
Статистическая обработка данных с использованием априорной информации	<a href="#">ссылка</a>	Статистическая обработка данных с использованием априорной информации	2.94%
Гибкий интеллектуальный интерфейс для систем передачи сложноорганизованной информации	<a href="#">ссылка</a>	Гибкий интеллектуальный интерфейс для систем передачи сложноорганизованной информации	2.55%

Рис. 6. Карта проекта для пользователя

Список проектов:

[Настройки системы](#)

[Назад](#)

Номер заявки	Название проекта	Статус заявки	
1	<a href="#">Аппаратно-программный комплекс для оптимизированного управления специализированными координатными столами лазерной и плазменной резки</a>	undone	<a href="#">Удалить</a>
30	<a href="#">первый проект</a>	undone	<a href="#">Удалить</a>

Рис. 7. Форма результатов работы системы для администратора

Данные проекта:

[Назад](#)

Номер заявки	Название проекта	Ключевые слова проекта	Аннотация проекта	Статус заявки
30	первый проект	методы интеллектуальной обработки информации	автоматизация анализа проектов с применением методов интеллектуальной обработки информации	done

Список аналогов (всего найдено 4033, из них отображено 10):

Название	Ссылка	Содержание	Степень совпадения	
Применение импульсных методов магнитного резонанса в устройствах обработки информации	<a href="#">ссылка</a>	Применение импульсных методов магнитного резонанса в устройствах обработки информации	4.86%	<a href="#">Подробнее</a>
Акустооптический эффект и его применение в системах оптической обработки информации	<a href="#">ссылка</a>	Акустооптический эффект и его применение в системах оптической обработки информации	4.2%	<a href="#">Подробнее</a>
"Психическая болезнь" как интеллектуальный проект	<a href="#">ссылка</a>	"Психическая болезнь" как интеллектуальный проект	3.59%	<a href="#">Подробнее</a>
Разработка и исследование системы обработки технологической информации гидрогенерирующих предприятий	<a href="#">ссылка</a>	Разработка и исследование системы обработки технологической информации гидрогенерирующих предприятий	3.59%	<a href="#">Подробнее</a>
Методы и средства обработки биоэлектрической информации	<a href="#">ссылка</a>	Методы и средства обработки биоэлектрической информации	3.59%	<a href="#">Подробнее</a>
Адаптивные алгоритмы обработки информации в мультиагентных системах	<a href="#">ссылка</a>	Адаптивные алгоритмы обработки информации в мультиагентных системах	3.59%	<a href="#">Подробнее</a>

Рис. 8. Карта проекта для администратора

Данные проекта:

[Назад](#)

Номер заявки	Название проекта	Ключевые слова проекта	Аннотация проекта	Статус заявки
30	первый проект	методы интеллектуальной обработки информации	автоматизация анализа проектов с применением методов интеллектуальной обработки информации	done

Данные аналога:

Название	Ссылка	Содержание	Степень совпадения
Применение импульсных методов магнитного резонанса в устройствах обработки информации	<a href="#">ссылка</a>	Применение импульсных методов магнитного резонанса в устройствах обработки информации	4.86%

Сравнение частотных характеристик проекта и аналога:

Частотная характеристика проекта		Частотная характеристика аналога	
интеллектуальной 2 информация 2 обработка 2	проект 2 метод 1 применение 1	импульсной 1 информация 1 магнитный 1 метод 1	обработка 1 применение 1 резонанс 1 устройство 1

Рис. 9. Результат нажатия на ссылку "Подробнее" для определенного аналога

работы системы, логины и пароли для доступа к таблицам базы данных, имя сервера СУБД (система работает на mysql) и названия самих таблиц. При этом, чтобы получить доступ к режиму администратора, нужно нажать на соответствующую ссылку в верхнем меню системы и пройти аутентификацию.

Карта проекта для администратора также более подробная (рис. 8). Кроме входных данных пользователя и отображения списка аналогов, число которых также можно регулировать в настройках системы (в данном случае оно равно 10), для каждого аналога после названия, ссылки, содержания и степени совпадения выводится ссылка "Подробнее". При нажатии на эту ссылку, кроме уже рассмотренных двух таблиц, отображается третья таблица, в которой приводится сравнение частотных характеристик входных данных (проекта) и выбранного аналога (рис. 9). Способы вывода могут быть настроены администратором системы (в данном случае термины выводятся в два столбца для проекта и аналога), при этом для проекта и для аналога отображаются термины в нормальной форме (для имени существительного, например, это именительный падеж, единственное число) и их частоты; совпадающие термины выделяются жирным шрифтом, и список терминов упорядочивается по убыванию частоты. Если термин находится в списке стоп-слов, он не отображается и не учитывается системой.

## Заключение

Система обладает некоторыми недостатками. Помимо упомянутых выше проблем, связанных с не всегда точной работой морфологического модуля, необходимостью обновления списка стоп-слов, главной проблемой системы является зависимость от ресурсов [5] и [6], хотя они и по-прежнему работают и несколько не изменились за многие годы. Еще одна сложность, которая возникла в результате тестирования системы, — необходимость разгружать сервер хостинга на определенное количество времени. Несмотря на то что сервер достаточно мощный, опытным путем было установлено, что главный скрипт системы может работать только 29 с (которые впоследствии были увеличены до 600 с), после чего должен давать "отдыхать" серверу, сбрасывая промежуточные данные и значения счетчиков циклов в базу данных или в отдельный файл, и возобновлять работу по очередному вызову планировщика cron. При превышении 29 (600) с либо могут исказиться данные, либо сервер автоматически останавливает скрипт, считая его "зависшим", несмотря на то что внутри скрипта есть команды отдыха (sleep) после каждой итерации главного цикла. При этом есть риск наложения работы одного и того же скрипта на самого себя, который необходимо исключать на 100 %. Дело в том, что настройки системы (в коде модуля поиска записано, что он должен работать 600 с с помощью команды `set_time_limit(600)`) и реальное время работы скрипта могут значительно отличаться — вплоть до 100 %, в связи с чем даже было проведено мини-исследование (основным выводом которого является интервал в заданиях планировщика (cron), равный аргументу функции `set_time_limit()`, умноженному на 2) со следующими результатами (приведен лишь фрагмент результатов):

*выставлено 300 секунд, по факту работает 459.973237991 секунд*

*выставлено 600 секунд, по факту работает 606.108923197 секунд*

*выставлено 600 секунд, по факту работает 774.679234982 секунд*

*выставлено 240 секунд, по факту работает 381.271329165 секунд*

*выставлено 1 секунд, по факту работает 1.03081297874 секунд*

*выставлено 600 секунд, по факту работает 680.900939941 секунд*

Еще одной проблемой системы является время обработки очередного "заказа" пользователя.

Чем больше текст, вводимый пользователем, тем дольше работает система, так как число терминов растет. В ходе тестирования и опытной эксплуатации системы было принято решение искать аналоги, анализируя только названия или темы документов (диссертаций), иначе работа системы может затянуться с учетом вышеописанных сложностей и тонкостей на несколько недель, что будет неприемлемо для пользователя. В зависимости от сложности заказа и серверных настроек системы результаты работы системы приходят на почту пользователя через промежуток времени в среднем от 20 мин до нескольких дней после успешного заполнения полей. Такой временной промежуток обусловлен необходимостью не загружать сервер слишком сильно и не перегружать ресурс [6] поисковыми запросами, а также очередью заявок системы.

Тестирование системы показало, что ресурс [6], несмотря на заявленный морфологический поиск на самом деле предоставляет его не всегда. Достаточно трудно определить, каковы же условия для входных данных, чтобы морфологический поиск учитывался ресурсом полностью; эксперименты выявили, что при вводе в систему ресурса словоформы "интеллектуальный" ресурс выводит одни результаты, а при вводе другой словоформы этого же слова — "интеллектуальной" — ресурс выводит другие результаты, которые пересекаются с первыми, но совсем не на 100 %. Другими словами, то, что есть в первой выборке, отсутствует во второй и наоборот. Это приводит к тому, что разработанная система должна учитывать не только нормальную форму слова ("интеллектуальный"), но и ту форму, которая фактически была использована при вводе входных данных ("интеллектуальной") и, следовательно, работает дольше, чем могла бы. Кроме того, система должна учитывать все словоформы для каждого слова. Вместе с этим тестирование системы выявило различие в кодировках разных почтовых сервисов (hse.ru — UTF-8, mail.ru — KOI8-U и т. д.).

С точки зрения практической полезности система за счет обработки любой информации о предстоящем исследовании позволяет получать полезную и иногда неожиданную информацию, которую нельзя получить с помощью традиционных систем поиска в каталогах диссертаций, так как у пользователя просто не хватает терпения и времени просмотреть и проверить все возможные варианты и комбинации терминов. Кроме того, система обладает максимально простым интерфейсом и выполняет всю работу автоматически, пользователю достаточно ввести входные данные и прочитать через некоторое время письмо со ссылкой

на результаты работы системы. Система доступна по адресу [9].

#### Список литературы

1. **Использование phpMorphy** — описание и методы. sourceforge / heromantor, последнее изменение от 14.01.2010. URL: <http://phpmorphy.sourceforge.net/dokuwiki/manual> (дата обращения: 24.12.2017).

2. **Сокирко А. В.** Морфологические модули на сайте [www.aot.ru](http://www.aot.ru). URL: <http://www.aot.ru/docs/sokirko/Dialog2004.htm> (дата обращения: 24.12.2017).

3. **Моченов С. В., Бледнов А. М., Луговских Ю. А.** Векторная модель представления текстовой информации: Материалы Международной научной конференции "Современные информационные технологии и письменное наследие: от древних рукописей к электронным текстам". URL: <http://mns.udsu.ru/conf/report/Mochenov2.pdf> (дата обращения: 24.12.2017).

4. **Соколов Е.** Метрические методы классификации (семинары). URL: [http://www.machinelearning.ru/wiki/images/9/9a/Sem1\\_knn.pdf](http://www.machinelearning.ru/wiki/images/9/9a/Sem1_knn.pdf) (дата обращения: 24.12.2017).

5. **Научная** электронная библиотека диссертаций и авторефератов disserCat. URL: <http://www.dissercat.com> (дата обращения: 15.12.2017).

6. **Электронная** библиотека Российской государственной библиотеки. URL: <http://sigla.rsl.ru> (дата обращения: 14.12.2017).

7. **phpMorphy**. URL: <http://sourceforge.net/projects/phpmorphy/> (дата обращения: 14.12.2017).

8. **Макаров С. Л.** Интеллектуальная система автоматизированного поиска диссертаций // Материалы V Всероссийской конференции студентов, аспирантов и молодых ученых "Искусственный интеллект: философия, методология, инновации", г. Москва, МГТУ МИРЭА, 9—11 ноября 2011 г. М.: Радио и Связь, 2011. 272 с.

9. **Система** STAPTLite. URL: [serjmak.com/startlite/](http://serjmak.com/startlite/) (дата обращения: 15.12.2017).

S. L. Makarov, Ph. D, Associate Professor at the School of Software Engineering  
of the Faculty of Computer Science, smakarov@hse.ru  
National Research University Higher School of Economics

## Information Technologies of a Search for Similar Researches in Dissertations Catalogues

*The article represents some information technologies of a search for similar researches, projects or just ideas in catalogues of dissertations and suggests a software web-based system of automated search of the kind. The relevance of the system is justified by impossibility of doing the same work by people and absence of systems which can input a large text for this kind of search. There are several methods described which the system is based upon as well as several mini-researches results which have been conducted in order to understand structures of the dissertations catalogues and to interpret results of the system's work. The system's architecture, database structure and algorithm are presented in the article along with the rules of its running automatically using one of the planning tasks mechanisms of operating systems (cron). There are several user's and the system administrator's interface examples and system outputs provided from the user's and the system administrator's points of view. In the conclusion, key features and difficulties of the system and of the process of using it are outlined together with the outcome of the research with advantages and disadvantages of the system along with mentioning its practical value. Also, as a part of the conclusion there are results of another mini-research about the system's search script lifetime and some outcomes of server and the script settings necessary for getting correct results with the help of the system.*

**Keywords:** information technologies, automated search, automation, software engineering, search system, project analysis, research summary, dissertations catalogue, user interface

#### References

1. **Ispol'zovaniye phpMorphy** — opisaniye i metody (phpMorphy Manual — Description and Methods), available at: <http://phpmorphy.sourceforge.net/dokuwiki/manual> (date of access: 12.24.2017) (in Russian).

2. **Sokirko A. V.** *Morfologicheskiye moduli na saite www.aot.ru* (Morphology Modules on www.aot.ru Site), available at: <http://www.aot.ru/docs/sokirko/Dialog2004.htm> (date of access: 12.24.2017) (in Russian).

3. **Mochenov S. V., Blednov A. M., Lugovskiy Yu. A.** *Vektornaya model' predstavleniya tekstovoy informatsiyi* (Vector Model for Representing Textual Information), *Materialy mezhdunarodnoy nauchnoy konferentsii "Sovremenniy informatsionnyye tehnologii I pis'mennoye naslediy: ot drevnih rukopisey k elektronnim tekstam"*, available at: <http://mns.udsu.ru/conf/report/Mochenov2.pdf> (date of access: 12.24.2017) (in Russian).

4. **Sokolov Ye.** *Metricheskiye metody klassifikatsii (seminary)* (Metric methods of classification (seminars)), available at: [http://](http://www.machinelearning.ru/wiki/images/9/9a/Sem1_knn.pdf)

[www.machinelearning.ru/wiki/images/9/9a/Sem1\\_knn.pdf](http://www.machinelearning.ru/wiki/images/9/9a/Sem1_knn.pdf) (date of access: 12.24.2017) (in Russian).

5. **Nauchnaya elektronnyaya biblioteka dissertatsiy I avtoreferatov disserCat** (disserCat Scientific Electronic Library of Dissertations and Their Abstracts), available at: <http://www.dissercat.com> (date of access: 12.15.2017) (in Russian).

6. **Elektronnyaya biblioteka Rossiyskoy gosudarstvennoy biblioteki** (Electronic Library of the Russian State Library), available at: <http://sigla.rsl.ru> (date of access: 12.14.2017) (in Russian).

7. **phpMorphy**, available at: <http://sourceforge.net/projects/phpmorphy/> (date of access: 12.14.2017).

8. **Makarov S. L.** *Intellektualnaya sistema avtomatizirovannogo poiska dissertatsiy* (Intellectual System for Automated Search for Dissertations), *Materialy V Vserossiyskoy Konferentsii Studentov, Aspirantov I Molodyh Uchyonih "Iskustvenniy Intellekt: Filosofiya, Metodologiya, Innovatsiyi"*, Moscow, Moscow Technological University (MIREA), 9—11 of November, 2011, Moscow, Radio e Svyaz, 2011, 272 p., pp. 202—204 (in Russian).

9. **StartLite** System, available at: [serjmak.com/startlite/](http://serjmak.com/startlite/) (date of access: 12.15.2017) (in Russian).

**А. Г. Шмелева**, канд. физ.-мат. наук, доц., e-mail: shmeleva\_a@mirea.ru,  
**А. И. Ладынин**, аспирант, преподаватель, e-mail: andrey.ladyinin@hotmail.ru,  
**А. В. Бахметьев**, студент, e-mail: a\_bahmetev@mail.ru,  
 Московский технологический университет, Москва

## **Построение взвешенных решений управления сложными производственными системами с применением теории массового обслуживания**

*Представлена интеллектуальная система (СППР "ШАГ") для поддержки принятия решений в задачах управления производством, включающая актуальные методы обработки и анализа информации. Одним из модулей представленной интеллектуальной системы является программная реализация моделей теории массового обслуживания, позволяющих проводить оценку пропускной способности и отказоустойчивости промышленных систем.*

**Ключевые слова:** система поддержки принятия решений, автоматизация, промышленное производство, имитационное моделирование, система массового обслуживания, программная реализация, пропускная способность

### **Введение**

Одной из проблем современной науки является разработка и внедрение в практику методов исследования функционирования сложных систем. К классу сложных систем относят технологические, производственные, энергетические комплексы, системы автоматизации управления и другие объекты. Механизмы эффективного управления производством требуют внедрения информационных технологий для принятия стратегических решений экономического и научно-технического развития. Имитационное моделирование является одним из наиболее мощных средств исследования разнообразных систем.

Имитационное моделирование направлено на построение взвешенных решений управления сложными производственными системами с помощью математических методов и алгоритмов [1, 2]. Создание систем поддержки принятия решений (СППР), где сложность модели можно сколь угодно увеличивать, чтобы добиться все более точного соответствия исследуемой системе, является актуальной задачей. СППР позволяет проводить многократные расчеты, изменяя параметры моделей, анализировать результаты и выбирать приоритетные альтернативы развития.

### **1. Проектирование информационных систем поддержки принятия производственных решений**

Цель управления высокотехнологичным предприятием — приведение объекта управления (системы) в желаемое состояние в усло-

виях действия внешних и внутренних возмущающих факторов, характеризующих состояние среды. Возможный способ достижения поставленных целей — анализ вариантов решений (альтернатив) [3—5]. Создание информационной системы направлено на выбор оптимальной стратегии развития на основе качественной оценки альтернатив по одному или нескольким параметрам. В общей постановке задача принятия решений заключается в выборе наиболее приоритетной альтернативы в соответствии с предпочтениями лица, принимающего решения (ЛПР).

К особенностям решения задач управления можно отнести:

- структуризацию входных данных;
- определение цели (предпочтения);
- выбор ограничений;
- описание элементов системы, не имеющих однозначной объективной оценки;
- математическое моделирование с применением совокупности методов оптимизации и планирования;
- разработку алгоритмов, позволяющих снизить неопределенность, обусловленную спецификой или новизной тематики.

Обобщенная модель принятия решений прогнозирования развития может быть представлена в виде множества исследуемых подмножеств параметров:

$$C_{opt} = \{P_0, T, R, P, G, L, C, K\},$$

где  $C_{opt}$  — развитие ситуации в результате принятого решения;  $P_0$  (problem) — характеристики проблемы, требующие анализа;  $T$  (time) — временные параметры;  $R$  (resources) — доступ-

ные ресурсы;  $P = (P_1, \dots, P_n)$  — альтернативные состояния системы, уточняющие исходную проблему;  $G$  (goals) — желаемые цели;  $L$  (limitations) — ограничения;  $C$  (choices) — альтернативные варианты решения;  $K$  — критерии, определяющие наилучшее решение.

Агрегации указанных множеств значимых параметров модели позволяет формализовать задачу принятия решений посредством выбора математических методов и алгоритмов.

Задача принятия решений требует тщательного исследования, обусловленного необходимостью однозначного определения, представления и формализации начальных данных, позволяющего применять имитационное моделирование исследуемых систем. На рис. 1 представлен процесс принятия решений в виде структурной схемы, отражающей основные этапы исследования систем управления.

Модель наглядно демонстрирует необходимость комплексного подхода к исследованию входных данных, закономерностей, методов, алгоритмов и в дальнейшем к анализу возможных альтернатив — выбору наилучшей стратегии управления. Одним из перспективных направлений исследований в области структуризации и алгоритмизации поиска решения является разработка систем поддержки принятия решений (СППР) [6—9]. Модули системы должны объединять актуальные методы обработки и анализа информации, позволяя лицу, принимающему решения, получить взвешенное заключение, подкрепленное результатами анализа альтернатив [10, 11].

При создании СППР необходимо разрабатывать методы обработки начальных данных, позволяющие агрегировать элементы исследуемой системы для применения математических моделей, создавать алгоритмы анализа и мониторинга этапов реализации проектов наукоемких производств. Применение СППР позволит выбирать оптимальную стратегию развития, проводить объективную оценку проектов, анализировать параметры, разрабатывать календарные планы выполнения работ.

Сформулируем требования, предъявляемые к СППР, обусловленные спецификой задач управления и потребностями пользователей:

- достаточность (полнота);
- актуальность;
- адекватность (достоверность, точность);
- защищенность;
- масштабируемость;
- интуитивность.

При разработке СППР необходимо учитывать специфику областей применения и ори-



Рис. 1. Модель принятия производственных решений

ентированность на пользователей. Интуитивность СППР является одним из аспектов востребованности — это накладывает ограничения на использование математических методов. Как показывает практика, искусственное завышение набора переменных, определяющих соответствие модели прототипу, сопровождается вводом дополнительных параметров и ограничений, что делает модели тяжелыми в восприятии, а следовательно, либо ограничивает число пользователей, так как требуется достаточный уровень подготовки, либо не позволяет выявить главные влияющие факторы.

Внедрение информационных систем на предприятиях не должно ставить задачу длительного обучения персонала математическим методам и программированию. СППР должны быть доступны широкому кругу пользователей и направлены на решение повседневных производственных задач. Пользователю должна быть предоставлена возможность выбора программных модулей, содержащих как базовые, так и усложненные математические модели.

Предприятия и корпорации заинтересованы в актуальных методах решения задач анализа и планирования с применением СППР, что в контексте импортозамещения ставит вопрос о разработке и внедрении российских интеллектуальных систем поддержки принятия решений, направленных на модернизацию и внедрение новых технологий производства.

Для поддержки принятия решений в задачах управления производством разработана интеллектуальная система (СППР "ШАГ"), включающая актуальные методы обработки и анализа информации. На рис. 2 (см. вторую сторону обложки) представлен интерфейс программного комплекса. Проектируемая информационная система имеет модульную структуру и позволяет ЛПР использовать методы, необходимые для анализа разнообразных производственных задач. Функциональные возможности СППР обеспечивают систематизированный подход к исследованиям и позволяют:

- обрабатывать слабоструктурированную информацию;

- выявлять и анализировать альтернативы;
- определять качественные и количественные характеристики систем;
- строить формализованные выводы на основе экспертных мнений;
- составлять календарные планы работ;
- строить прогностические модели;
- выявлять значимые факторы возникновения нештатных ситуаций;
- решать задачи оптимизации.

СППР "ШАГ" направлена на проведение модельных экспериментов, которые невозможно или сложно провести на реальном объекте, а также предоставляет возможность получения новых знаний об исследуемых системах. Целью является усовершенствование методов управления, проектирования и анализа производственных процессов [6].

## 2. Теоретические предпосылки и постановка задач оценки пропускной способности и отказоустойчивости промышленных систем

Одним из модулей представленной интеллектуальной системы является программная реализация моделей теории массового обслуживания (ТМО), позволяющих проводить оценку пропускной способности и отказоустойчивости промышленных систем.

Модуль ориентирован на ускорение, удешевление, упрощение решения задач транспортной логистики, расчета числа персонала (операторов, диспетчеров), проектирования пропускной способности цеха и других задач теории массового обслуживания, возникающих на высокотехнологичном предприятии.

К основным элементам систем массового обслуживания (СМО) относят [12, 13]:

- входящий поток заявок;
  - аппараты обслуживания;
  - очередь;
  - исходящий поток обслуженных заявок.
- В литературе принято выделять три типа СМО:
- системы с отказами, для которых заявки, не обеспеченные свободным исполнителем, теряются;
  - системы с ожиданием, в которых присутствует накопитель заявок бесконечной емкости, формирующий очередь на обслуживание;
  - системы с накопителями конечной емкости (ожиданием и ограничениями), в которых длина очереди не может превышать ее максимальных значений.

Приоритетными являются СМО третьего типа, практическая значимость которых пред-

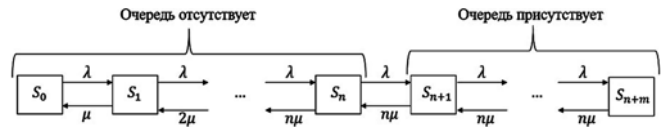


Рис. 3. Граф СМО с ограниченной очередью

ставляет наибольшую ценность для решения прикладных задач. Рассмотрим математическую модель и определяющие соотношения более подробно.

В систему, имеющую  $n$  обслуживающих приборов, поступает на вход поток заявок с интенсивностью  $\lambda$ . Интенсивность выполнения каждым каналом заявки равна  $\mu$ , а максимальное число мест в очереди —  $m$  (рис. 3).

Выделим три основных состояния системы:  $S_0$  — все аппараты обслуживания свободны, как и очередь;

$S_l$  — заняты  $l$  аппаратов обслуживания ( $l = \overline{1, n}$ ), очереди нет;

$S_{n+i}$  — заняты все  $n$  аппаратов обслуживания, в очереди пребывает  $i$  заявок ( $i = \overline{1, m}$ ).

Отметим, что основными параметрами, характеризующими СМО, являются ее пропускная способность и отказоустойчивость. Проведем исследование СМО третьего типа с использованием разработанной прикладной программы. Для этого, применяя следующие определяющие соотношения, решим практическую задачу.

Найдем предельную вероятность состояний согласно следующей формуле:

$$p_0 = \left( 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \dots + \frac{\rho^{n+m}}{n^m n!} \right)^{-1} = \left( 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \dots + \frac{\rho^{n+1}}{n n!} \frac{1 - \left(\frac{\rho}{n}\right)^m}{1 - \frac{\rho}{n}} \right), \quad (1)$$

где  $\rho = \lambda/\mu$  — коэффициент загрузки СМО.

Сумма вероятностей  $p_n, p_{n+1}, \dots, p_{n+m-1}$  есть вероятность образования очереди  $p_{оч}$ :

$$p_{оч} = \sum_{i=0}^{m-1} p_{n+i} = \frac{\rho^n}{n!} \frac{1 - \left(\frac{\rho}{n}\right)^m}{1 - \frac{\rho}{n}} p_0. \quad (2)$$

Заявка получает отказ в обслуживании, когда в очереди заняты все места:

$$p_{отк} = p_{n+m} = \frac{\rho^{n+m}}{n^m n!} p_0. \quad (3)$$

Относительная пропускная способность системы

$$Q = p_{\text{обс}} = 1 - p_{\text{отк}} = 1 - \frac{\rho^{n+m}}{n^m n!} p_0. \quad (4)$$

Абсолютная пропускная способность  $A$  определяется следующим образом:

$$A = \lambda Q = \lambda \left( 1 - \frac{\rho^{n+m}}{n^m n!} p_0 \right). \quad (5)$$

Среднее число заявок, пребывающих в очереди

$$L_{\text{оч}} = \sum_{i=1}^m i p_{n+i} = \frac{\rho^{n+1}}{n n!} \frac{1 - \left(\frac{\rho}{n}\right)^m \left[ 1 + m \left(1 - \frac{\rho}{n}\right) \right]}{\left(1 - \frac{\rho}{n}\right)^2} p_0. \quad (6)$$

Среднее число заявок, находящихся на исполнении в системе

$$L_{\text{обс}} = \frac{A}{\mu} = \rho \left( 1 - \frac{\rho^{n+m}}{n^m n!} p_0 \right). \quad (7)$$

Среднее число заявок, пребывающих в системе

$$L_{\text{СМО}} = L_{\text{оч}} + L_{\text{обс}}. \quad (8)$$

Среднее время пребывания заявки в СМО

$$T_{\text{СМО}} = \frac{L_{\text{СМО}}}{\lambda} = \frac{L_{\text{оч}}}{\lambda} + \frac{Q}{\mu}. \quad (9)$$

Среднее время нахождения заявки в очереди

$$T_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda}. \quad (10)$$

### 3. Решение задачи оценки параметров производственного процесса

Рассмотрим пример использования разработанного программного модуля для решения задачи оценки параметров производственного процесса — пропускной способности автоматизированного участка. Представим технологическое оборудование в виде СМО — совокупности заявок (однородных задач) и аппаратов обслуживания (станков), выполняющих однотипные действия.

Представим технологический процесс обработки деталей, используя аппарат ТМО. Допустим, ожидаемая загрузка производственной линии составляет 220 деталей в течение восьмичасовой рабочей смены. Число станков равно 8, среднее время обработки каждой заготовки на станке — 12 мин, длина очереди заявок на обслуживание не должна превышать 4. При этом, если заявка не попадает на обслуживание или

в очередь, она теряется. Решим задачу оценки пропускной способности и определим возможность использования данного технологического оборудования.

На основе представленных выше определяющих соотношений сформулируем задачу с использованием аппарата ТМО. По условию дана СМО с восемью каналами обслуживания, ожиданием и ограниченной очередью в четыре заявки. Дано, что в течение 8 ч поступает 220 заявок, что позволяет рассчитать интенсивность их поступления:

$$\lambda = \frac{N}{n} = \frac{220}{8 \cdot 60} = 0,458 \text{ 1/мин},$$

где  $N$  — число заявок, поступающих за исследуемое время работы;  $n$  — число станков.

В свою очередь, интенсивность обслуживания одной заявки составляет

$$\mu = \frac{1}{T_{\text{обс}}} = \frac{1}{12} = 0,083 \text{ 1/мин},$$

где  $T_{\text{обс}}$  — время обслуживания одной заявки.

Представим начальные параметры технологической системы:

число каналов обслуживания  $n = 8$ ;

интенсивность поступления заявок  $\lambda \approx 0,46$  1/мин;

интенсивность обслуживания заявки каждым из станков  $\mu \approx 0,08$  1/мин;

максимальная длина очереди  $m = 4$ .

Рассмотрим решение представленной задачи с использованием разработанного прикладного программного модуля. Для этого в диалоговом окне выберем пункт меню, соответствующий данной модели — многоканальной СМО с ограниченной очередью и отказами. Введем начальные данные задачи и перейдем к расчету выходных параметров СМО (рис. 4, 5).

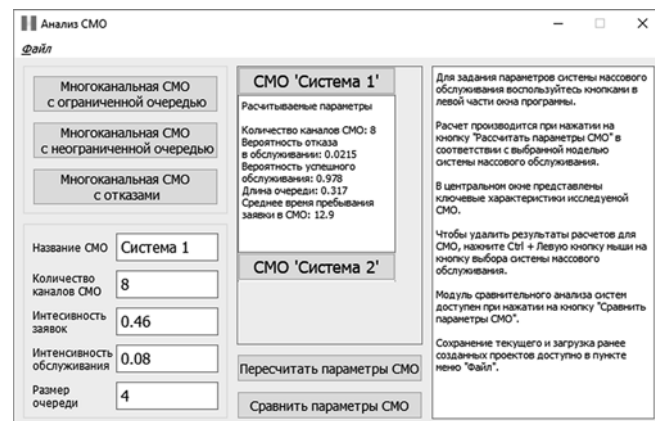


Рис. 4. Главное окно программного модуля



Модуль сравнения

Добавить СМО СМО 'Система 2' Экспортировать

	СМО с ограниченной очередью	СМО с неограниченной очередью	СМО с отказами					
	Каналы обслуживания	Интенсивность заявок	Интенсивность обслуживания	Размер очереди	Вероятность отказа	Вероятность обслуживания	Заявок в очереди	Среднее время ожидания
СМО 'Система 1'	8	0,46	0,08	4	0,0215	0,978	0,317	12,9
СМО 'Система 2'	8	1,04	0,08	4	0,272	0,728	1,86	10,9

Рис. 5. Результаты расчетов

В результате расчетов выходные параметры оцениваемой производственной системы составляют следующие значения: вероятность успешной обработки заготовки составляет 97,8 %; среднее число заявок в очереди равняется 0,32; среднее время пребывания заявки на обслуживании — 13 мин. Полученные данные свидетельствуют об избыточных производственных возможностях оборудования, рассматриваемая технологическая система способна обеспечить достаточные показатели отказоустойчивости.

На рис. 5 представлены результаты моделирования исходной системы с заданными начальными параметрами и идентичной системы, в которой среднее число заготовок, поступающих в течение рабочей смены, равняется 500 шт.

Во втором случае вероятность обслуживания составит 74,2 %; среднее число заявок в очереди — 1,9; среднее время ожидания — 11 мин. Полученные результаты свидетельствуют о возможности увеличения потока деталей, требующих обработки, однако повышение загрузки оборудования влечет увеличение вероятности технологических сбоев. Результаты моделирования позволяют сделать вывод о степени загруженности технологического оборудования, оценить пропускную способность совместно с допустимым числом отказов в обслуживании.

Представленный программный модуль, в основе которого лежат алгоритмы теории массового обслуживания, может быть использован для оценки параметров производственных систем на этапах проектирования и ввода в эксплуатацию, в ходе переоснащения предприятий и модернизации технологического оборудования. Приведенный пример свидетельствует о необходимости интеграции методов имитационного моделирования, позволяющих проводить исследование сложных систем с использованием прикладных программ.

### Заключение

Разработка информационных систем является актуальным направлением исследований мировых научных сообществ. На сегодняшний

день информационные системы должны соответствовать не только накоплению и хранению данных, но иметь направленность на помощь пользователю в принятии взвешенных решений на основе анализа имеющейся информации. Конкурентоспособность предприятия тесно связана с научно-техническим прогрессом, а следова-

тельно, с наличием технологий, позволяющих эффективно обрабатывать массивы данных для построения комплексных моделей анализа информационных потоков.

В статье представлена модель информационной системы, на основе которой разработана СППР "ШАГ", реализующая алгоритмы имитационного моделирования производственных систем. Показаны некоторые возможности модуля решения задач с применением теории массового обслуживания, проведены расчеты для оценки производственных систем. Использование предлагаемой программной реализации позволит усовершенствовать проектирование и анализ производственных процессов.

### Список литературы

1. Kashirskaya E. N., Kurnasov E. V., Kholopov V. A., Shmeleva A. G. Methodology for assessing the implementation of the production process // Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS). IEEE, 2017, P. 232—235. DOI: 10.1109/CTS.2017.8109533.
2. Kashirskaya E. N., Kholopov V. A., Shmeleva A. G., Kurnasov E. V. Simulation model for monitoring the execution of technological processes // Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS). IEEE, 2017, P. 307—310. DOI: 10.1109/CTS.2017.8109533.
3. Смирнова Э. Е., Таланова Ю. В., Шмелева А. Г. Эффективные методы управления проектами транспортной отрасли // Качество. Инновации. Образование. 2016. № 8—10 (135—137). С. 225—230.
4. Шмелева А. Г., Ладынин А. И. Система поддержки принятия решений "ШАГ" (СППР "ШАГ") // Свидетельство о государственной регистрации программ для ЭВМ РОСПАТЕНТ № 2017619109 от 15.08.2017 г.
5. Шмелева А. Г., Ладынин А. И., Бахметьев А. В. Некоторые аспекты формирования стратегических решений развития предприятия // Международная научно-техническая конференция "Информатика и технологии. Инновационные технологии в промышленности и информатике" ("МНТК ФТИ-2017") Сборник научных трудов под ред. Булатова М. Ф., 2017. С. 325—327.
6. Холопов В. А., Ладынин А. И. Анализ структур по отношению к типам производства // Промышленные АСУ и контроллеры. 2015. № 6. С. 7—11.
7. Kar A. K. A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network // Journal of Computational Science. 2015. N. 6. P. 23—33.
8. Saha C., F. Aqlan F., Lam S. S., Boldrin W. A decision support system for real-time order management in a heterogeneous production environment // Expert Systems with Applications. 2016. Vol. 60. P. 16—26.

9. Li B., Li J., Li W., Shirodkar S. A. Demand forecasting for production planning decision-making based on the new optimised fuzzy short time-series clustering // *Production Planning & Control*. 2012. Vol. 23(9). P. 663–673.

10. Abraham A. Intelligent Systems: Architectures and Perspectives, Recent Advances in Intelligent Paradigms and Applications, in *Studies in Fuzziness and Soft Computing*, Chapter 1, (eds A., Abraham, L., Jain and J., Kacprzyk) // Springer Verlag, Germany. 2002. P. 1–35.

11. Andreadis G., Klazoglou P., Niotaki K., Bouzakis K.-D. Classification and review of multi-agents systems in the manufacturing section // *Procedia Engineering*. 2014. Vol. 69. P. 282–290.

12. Дудин С. А. Системы массового обслуживания с сессионным поступлением запросов. Минск: РИВШ, 2011. 130 с.

13. Гнеденко Б. В., Коваленко И. Н. Введение в теорию массового обслуживания. Изд. 6. М.: URSS, 2013. 400 с.

A. G. Shmeleva, PhD, Associate Professor, e-mail: shmeleva\_a@mirea.ru,

A. I. Ladynin, Postgraduate Student, Lecturer, e-mail: andrey.ladynin@hotmail.ru,

A. V. Bakhmetiev, Student, e-mail: a\_bahmetev@mail.ru,

Moscow Technological University

## Weighted Decisions Development for Complex Production Systems Management Using the Theory of Mass Service

*The article presents intellectual system (DSS “ShAG”) which is intended to support decision-making in production management tasks; it includes up-to-date methods for information processing and analysis. The Decision Support Systems development process implies taking into account application areas and user focus. The presented model visually demonstrates necessity for a comprehensive approach regarding input data, patterns, methods, algorithms for choosing the best production management strategy studies. The following article presents developed program realization, which is consistent with commonly used corporate and resource planning standards — Manufacturing Resource Planning (MRP) and Enterprise Resource Planning (ERP). These methods and models, presented in the following DSS, could be successfully used in planning tasks for small businesses and corporations. DSS modules should combine relevant processing and analyzing information methods, allowing the decision-maker receive a weighted conclusion supported by the results of the alternatives’ analysis. One of the modules of the presented intellectual system is the models of queuing theory software implementation, which allows the throughput and fault tolerance of industrial systems assessment. The module is aimed at speeding up, cost lowering, simplifying the solution of transport logistics tasks, calculating the number of personnel (operators, dispatchers), designing the workshop throughput and other queuing theory problems that arise in high-tech enterprises. Developed software module usage example is presented for the production process parameters estimating problem. Usage of the proposed software will improve the engineering and production processes analysis.*

**Keywords:** decision support system, automation, stream production, simulation modeling, queuing system, business planning, software implementation, intellectual system, information analysis, enterprise resource planning, algorithms, throughput analysis

### References

1. Kashirskaya E. N., Kurnasov E. V., Kholopov V. A., Shmeleva A. G. Methodology for assessing the implementation of the production process, *Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS)*. IEEE, 2017, pp. 232–235. DOI: 10.1109/CTS.2017.8109533.

2. Kashirskaya E. N., Kholopov V. A., Shmeleva A. G., Kurnasov E. V. Simulation model for monitoring the execution of technological processes, *Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS)*. IEEE, 2017, pp. 307–310. DOI: 10.1109/CTS.2017.8109553.

3. Smirnova E. E., Talanova Yu. V., Shmeleva A. G. Transport industry effective project management methods, *Kachestvo. Innovacii. Obrazovanie* (Quality. Innovation. Education), 2016. no. 8–10 (135–137), pp. 225–230 (in Russian).

4. Shmeleva A. G., Ladynin A. I. Decision Support System ShAG (DSS ShAG). Certificate of state registration of computer programs ROSPATENT No. 2017619109, August 15, 2017.

5. Shmeleva A. G., Ladynin A. I., Bakhmetev A. V. Some aspects of enterprise development strategic solutions forming, *Mezhdunarodnaja nauchno-tehnicheskaja konferencija “Informatika i tehnologii. Innovacionnye tehnologii v promyshlennosti i informatike” (“MNTK FTI-2017”) Sbornik nauchnyh trudov*. Ed. Bulatov M. F. (In the collection: International Scientific and Technical Conference “Informatics and Technology. Innovative technologies in industry and informatics (“MNTK FTI-2017”) Collection of scientific works. Ed. by Bulatov M. F.), 2017, pp. 325–327 (in Russian).

6. Kholopov V. A., Ladynin A. I. ACS Analysis on the Relation of Different Manufacturing Types, *Promyshlennye ASU i kontroly* (Industrial Automated Control Systems and controllers), 2015. no. 6, pp. 7–11 (in Russian).

7. Kar A. K. A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network, *Journal of Computational Science*, 2015, no. 6, pp. 23–33.

8. Saha C., Aqlan F., Lam S. S., Boldrin W. A decision support system for real-time order management in a heterogeneous production environment. *Expert Systems with Applications*, 2016, 60, pp. 16–26.

9. Li B., Li J., Li W., and Shirodkar S. A. Demand forecasting for production planning decision-making based on the new optimised fuzzy short time-series clustering, *Production Planning & Control*, 2012, 23 (9), pp. 663–673.

10. Abraham A. Intelligent Systems: Architectures and Perspectives, Recent Advances in Intelligent Paradigms and Applications, in *Studies in Fuzziness and Soft Computing*, Chapter 1, (eds A., Abraham, L., Jain and J., Kacprzyk), Springer Verlag, Germany, 2002, pp. 1–35.

11. Andreadis G., Klazoglou P., Niotaki K., Bouzakis K.-D. Classification and review of multi-agents systems in the manufacturing section, *Procedia Engineering*, 2014, vol. 69, pp. 282–290.

12. Дудин С. А. Системы массового обслуживания с сессионным поступлением запросов (Queuing systems with session incoming requests), Минск, РИВШ Publ., 2011, 130 p. (in Russian).

13. Гнеденко Б. В., Коваленко И. Н. (Vvedenie v teoriiu massovogo obsluzhivaniya) Introduction to the theory of queuing, Izd. 6. Moscow, URSS Publ., 2013, 400 p. (in Russian).

**Л. Е. Назаров**, д-р физ.-мат. наук, вед. науч. сотр., e-mail: nazarov@ire.rssi.ru, Фрязинский филиал Института радиотехники и электроники им. В. А. Котельникова РАН, г. Фрязино  
**П. В. Шишкин**, нач. сектора, e-mail: p\_v\_s@bk.ru,  
АО "Информационные спутниковые системы" им. академика М. Ф. Решетнева, г. Железногорск

### Характеристики помехоустойчивых блочных турбокодов на основе низкоплотных кодов

*Приведены результаты исследований характеристик блочных турбокодов, формируемых с использованием составляющих низкоплотных кодов на основе конечной Евклидовой геометрии. Показано, что при приеме этих турбокодов можно применить алгоритмы итеративного приема, разработанные для общего класса низкоплотных кодов. Даны результаты моделирования алгоритмов итеративного приема для ряда рассматриваемых турбокодов.*

**Ключевые слова:** помехоустойчивые коды, турбокоды, низкоплотные коды, конечная Евклидова геометрия, алгоритмы итеративного приема

#### Введение

Помехоустойчивые коды используют в цифровых системах связи в целях повышения надежности передачи информации по радиоканалам [1]. Кодовые конструкции под общим названием блочные турбокоды рассматриваются как одни из наиболее перспективных относительно вероятностных характеристик и сложности алгоритмов приема [2]. Данные турбокоды являются базовыми для ряда протоколов передачи информации, например для протоколов IESS-15, IESS-315, IESS-316e, разработанных для спутниковой системы связи Intelsat.

Рассматриваемые турбокоды формируются на основе последовательного объединения составляющих блочных кодов [2]. Суть алгоритмов приема этих кодов — декомпозиция оптимального правила приема на совокупность этапов обработки, соответствующих составляющим кодам, что приводит к существенному упрощению результирующих процедур приема [2, 3].

Класс известных составляющих кодов ограничен — используются блочные коды Хэмминга, простые коды с обобщенной проверкой на четность [2–4]. Актуальной является проблема разработки и исследования характеристик блочных турбокодов на основе составляющих кодов из других классов с простыми алгоритмами приема. В работе рассматриваются турбокоды, формируемые с использованием составляющих низкоплотных кодов

на основе конечных геометрий [1, 5, 6]. Для этих кодов известны алгоритмы формирования и приема, характеризующиеся низкой сложностью реализации [1, 5–7]. Приведены алгоритмы итеративного приема данных кодовых конструкций и результаты их компьютерного моделирования при наличии канального аддитивного белого гауссовского шума (АБГШ).

#### 1. Постановка задачи

Блочный турбокод на основе составляющих двоичных кодов  $C_1$ ,  $C_2$  с параметрами  $(n_1, k_1)$  и  $(n_2, k_2)$  эквивалентен двумерной матрице [1, 2]. Строки матрицы — кодовые слова кода  $C_1$ , столбцы матрицы — кодовые слова кода  $C_2$ . Здесь  $n$ ,  $k$  — длительность и информационный объем кодовых слов. Длительность кодовых слов кода-произведения равна  $N = n_1 n_2$ , объем информационного блока  $K = k_1 k_2$ , скорость кода  $R = K/N$ .

Алгоритмы оптимального приема сигналов, соответствующих помехоустойчивым кодам, основаны на вычислении множества  $2^K$  корреляций входной реализации с этими сигналами [1, 6]. Для  $K \gg 1$  реализация алгоритмов оптимального приема представляет трудноразрешимую проблему.

Для рассматриваемых турбокодов, формируемых с использованием простых составляющих блочных кодов (коды Хэмминга, коды с обобщенной проверкой на четность), разработаны

алгоритмы итеративного приема [2–7]. Эти алгоритмы характеризуются существенно меньшей сложностью реализации при незначительных энергетических потерях по отношению к оптимальному приему. Итерация итеративного приема включает выполнение двух этапов. На первом этапе на основе отсчетов входной реализации и априорных вероятностей относительно кодовых символов вычисляются апостериорные вероятности приема для символов кода  $C_1$  (прием по горизонтали) [8]. Функционалы от вычисленных апостериорных вероятностей принимаются как априорные вероятности для кодовых символов, используемые на втором этапе итерации при вычислении апостериорных вероятностей символов кода  $C_2$  (прием по вертикали). После выполнения задаваемого числа итераций принимаются решения относительно кодовых символов турбокода с использованием вычисленных апостериорных символьных вероятностей.

Теоретический анализ и моделирование алгоритмов итеративного приема рассматриваемых турбокодов показали их эффективность относительно вероятностных характеристик по сравнению с известными схемами помехоустойчивого кодирования, включая сверточные коды в сочетании с алгоритмом приема Витерби [9]. При увеличении информационных объемов турбокодов достигаются вероятностные характеристики при приеме, близкие к предельным теоретическим характеристикам [5].

Для рассматриваемых турбокодов, формируемых с использованием составляющих низкоплотностных кодов, можно применить алгоритмы итеративного приема, разработанные для общего класса низкоплотностных кодов. Обоснование этого подхода основано на том факте, что эти турбокоды также входят в класс низкоплотностных кодов [1]. Ниже рассматриваются составляющие низкоплотностные коды на основе конечной Евклидовой геометрии [1, 10], которые входят в класс циклических кодов и характеризуются низкой сложностью алгоритмов формирования и итеративного приема.

Суть задачи — разработка и исследование вероятностных характеристик алгоритмов итеративного приема турбокодов, формируемых с использованием составляющих низкоплотностных кодов на основе конечной Евклидовой геометрии.

## 2. Низкоплотностные коды на основе конечной Евклидовой геометрии и блочные турбокоды на их основе

Пусть  $EG(m, 2^s)$  —  $m$ -мерная конечная Евклидова геометрия над полем  $GF(2^s)$ ,  $m, s$  — положительные целые числа [1]. Эта геометрия со-

держит  $2^{ms}$  точек в виде векторов с  $m$  компонентами поля  $GF(2^s)$ .  $EG(m, 2^s)$  содержит  $2^{(m-1)s} \times (2^{ms} - 1)/(2^s - 1)$  линий с  $2^s$  точками в своем составе.

Геометрию  $EG(m, 2^s)$  можно рассматривать как расширение поля  $GF(2^{ms})$  над  $GF(2^s)$  [1, 10]. Пусть  $\alpha$  — примитивный элемент этого поля, множество элементов  $0, \alpha^0, \alpha^1, \dots, \alpha^{2^{ms}-2}$  эквивалентно  $2^{ms}$  точкам геометрии  $EG(m, 2^s)$ , а  $\alpha^i, \alpha^j$  — линейно независимые точки в  $EG(m, 2^s)$ . Множество точек, задаваемых соотношениями  $\{\alpha^i + \beta\alpha^j; \beta \in GF(2^s)\}$ , определяет линию инцидентности, проходящую через  $\alpha^i$ . Пусть  $H_{EG}(m, s)$  — матрица с элементами из  $GF(2)$ , строки которой эквивалентны линиям инцидентности геометрии  $EG(m, 2^s)$ , не проходящим через 0. Эта матрица является проверочной матрицей низкоплотностного кода на основе геометрии  $EG(m, 2^s)$ . Матрица  $H_{EG}(m, s)$  содержит  $J = (2^{(m-1)s} - 1)/(2^s - 1)$  строк и  $n = 2^{ms} - 1$  столбцов, строки матрицы имеют вес Хэмминга  $J_N = 2^s$ , столбцы имеют вес Хэмминга  $J_D = (2^{ms} - 1)/(2^s - 1) - 1$ .

Низкоплотностные коды на основе Евклидовой геометрии  $EG(m, 2^s)$  являются циклическими кодами с порождающим многочленом  $g_{EG}(x)$  [1]. Длина кодовых слов равна  $n = 2^{ms} - 1$ , минимальный вес Хэмминга  $d_{\min} \geq J_D + 1$ , размерность  $k$  определяется степенью порождающего многочлена  $g_{EG}(x)$ . Пусть  $\alpha$  — примитивный элемент поля  $GF(2^{ms})$ , элемент  $\alpha^h$  является корнем  $g_{EG}(x)$  тогда и только тогда, когда выполняется условие [1]:

$$0 < \max_{0 \leq l < s} (W_{2^s}(h^{(l)})) \leq (m-1)(2^s - 1); \quad (1)$$

$$W_{2^s}(h) = \delta_0 + \delta_1 + \dots + \delta_{m-1}; \quad (2)$$

$$h = \delta_0 + \delta_1 2^s + \dots + \delta_{m-1} 2^{(m-1)s}, \quad (3)$$

$$0 \leq \delta_i < 2^s, 0 \leq i < m,$$

где  $0 \leq \delta_i < 2^s, 0 \leq i < m$  — коэффициенты при разложении (3) показателя степени  $h$  элемента  $\alpha$ ;  $h^{(l)}$  — остаток величины  $h2^l$  по модулю  $(2^{ms} - 1)$ .

В табл. 1 приведены параметры  $n, k, J_N, J_D$ , а также показатели степеней примитивного элемента  $\alpha$  для порождающих многочленов, полученные с использованием соотношений (1)–(3) для ряда кодов на основе Евклидовой

Таблица 1

Параметры низкоплотностных кодов  $EG(m, 2^s)$  ( $\alpha$  — примитивный элемент поля  $GF(2^{ms})$ )

$(m, 2^s)$	$n$	$k$	$d_{\min}$	$J_N$	$J_D$	Показатели степеней $\alpha^i$
$(2, 2^2)$	15	7	5	4	4	1, 3
$(2, 2^3)$	63	37	9	8	8	1, 3, 5, 7, 21
$(2, 2^4)$	255	175	17	16	16	1, 3, 5, 7, 9, 11, 13, 15, 37, 45

Таблица 2

Параметры блочных турбокодов, формируемых с использованием низкоплотных кодов  $EG(m, 2^s)$

$(m, 2^s)$	$n$	$k$	$d_{\min}$	$J_N$	$J_D$	$R = k/n$
$(2, 2^2)$	225	49	25	4	8	0,22
$(2, 2^3)$	3969	1369	81	8	16	0,34
$(2, 2^4)$	65 025	30 625	289	16	32	0,47

геометрии  $EG(m, 2^s)$  [7]. В табл. 2 приведены параметры  $n, k, J_N, J_D$  для блочных турбокодов на основе этого ряда низкоплотных кодов.

Приведем описание алгоритма итеративно-го приема ВР (*belief propagation*) низкоплотных кодов [5, 6, 10].

Обозначим  $H = (h_{ij}; 0 \leq l < n - k; 0 \leq i < n)$  проверочную матрицу низкоплотного кода с параметрами  $(n, k)$ ,  $n$  — длительность кодовых слов  $\mathbf{V} = (b_0, b_1, \dots, b_{n-1})$ ,  $k$  — размерность кода. Пусть  $\mathbf{Y} = (y_0, y_1, \dots, y_{n-1})$  — дискретная реализация с выхода демодулятора сигналов, поступающая на вход декодера, отсчеты реализации  $\mathbf{Y}$  задаются в виде  $y_i = s_i + n_i$ , где  $s_i$  — сигнальные составляющие;  $n_i$  — помеховые составляющие,  $i = 0, 1, \dots, n - 1$ . Введем обозначение  $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})$  — последовательность "жестких" решений, т.е.  $x_i = 0$  при условии  $y_i \geq 0$  и  $x_i = 1$  — в противном случае.

Рассматриваемые низкоплотные коды на основе конечных геометрий обладают свойством организации множества ортогональных проверочных соотношений для каждого кодового символа  $b_i$  кодовых слов  $\mathbf{V}$  [1, 6]. Пусть  $N(m) = (i: h_{mi} = 1)$  — множество номеров позиций кодовых символов объемом  $J_N(m)$ , образующих  $m$ -е проверочное соотношение;  $N(m)/l$  — множество  $N(m)$  без  $l$ -го символа;  $D(l) = (m: h_{ml} = 1)$  — множество проверочных ортогональных соотношений относительно кодового символа  $b_l$  объемом  $J_D(l)$ ;  $D(l)/m$  — множество ортогональных проверок  $D(l)$  без  $m$ -й проверки. Рассматриваемые низкоплотные коды являются регулярными, т.е. выполняются условия  $J_N(m) = J_N$  и  $J_D(l) = J_D$  для всех  $m, l$ .

Алгоритм итеративного приема ВР использует описание низкоплотных кодов в виде двуслойного графа Таннера, задаваемого проверочной матрицей  $H$  [11]. Вершины первого



Рис. 1. Вид графа Таннера для низкоплотного кода  $EG(2, 2^2)$  с параметрами  $J_N = J_D = 4$  ( $\oplus$  — сложение в поле  $GF(2)$ )

слоя графа соответствуют кодовым символам, вершины второго слоя — ортогональным проверочным соотношениям. На рис. 1 в качестве примера приведен вид графа, соответствующего низкоплотному коду  $(15, 7)$  с параметрами  $J_N = J_D = 4$ .

Алгоритм ВР включает инициализацию и три этапа обработки при выполнении итерации [1, 6, 10].

**Инициализация.** Устанавливаются начальные значения величин  $z_{mi} = y_i, m \in J_D(i); i = 0, 1, \dots, n - 1$ .

**Шаг 1.** Вычисляется последовательность "жестких" решений

$$\sigma_{mi} = \begin{cases} 1, & z_{mi} > 0, \\ 0, & z_{mi} \leq 0. \end{cases} \quad (4)$$

Для каждой ортогональной проверки  $m$  вычисляются величины  $\sigma_m, L_{mi}$

$$\sigma_m = \sum_{i \in N(m)} \sigma_{mi} \pmod{2}; \quad (5)$$

$$L_{mi} = (-1)^{\sigma_m \oplus \sigma_{mi} \oplus 1} \min_{i' \in N(m)/i} (|z_{mi'}|). \quad (6)$$

**Шаг 2.** На основе значений  $L_{mi}$  вычисляются величины  $z_{mi}$  для последующей итерации

$$z_{mi} = y_i + \sum_{m' \in D(i)/m} L_{m'i} \quad (7)$$

**Шаг 3.** При невыполнении требуемого числа итераций выполняется шаг 1 последующей итерации, иначе принимается решение относительно передаваемых кодовых символов  $b_i$  с использованием величин  $z_i$

$$z_i = y_i + \sum_{m \in D(i)} L_{mi}. \quad (8)$$

Принимается решение  $b_i = 0$ , если  $z_i \geq 0$ , иначе  $b_i = 1$ .

Приведенный алгоритм итеративного приема ВР осуществляет параллельное использование величин  $L_{mi}$  для вычисления значений  $z_{mi}$  при реализации соотношения (7), т.е. на шаге 1 вычисляется полное множество  $\{L_{mi}\}$  и после этого реализуется шаг 2. Модификация этого алгоритма ( $m$ -ВР) заключается в реализации последовательного использования величин  $L_{mi}$  при вычислении  $z_{mi}$ , т.е. шаг 2 реализуется после вычисления очередного значения  $L_{mi}, i = 0, 1, \dots, n - 1$ , не требуя вычисления полного множества  $\{L_{mi}\}$  [12].

Рассматриваемые турбокоды, формируемые с использованием низкоплотных кодов с параметрами  $J_N$  и  $J_D$ , входят в класс низкоплотных кодов с параметрами  $J_N$  и  $2J_D$  [1], поэтому при их приеме применяется приведенный алгоритм итеративного приема ВР (4)–(8).

### 3. Результаты моделирования

На рис. 2—4 приведены результаты моделирования алгоритмов итеративного приема для ряда рассматриваемых блоковых турбокодов

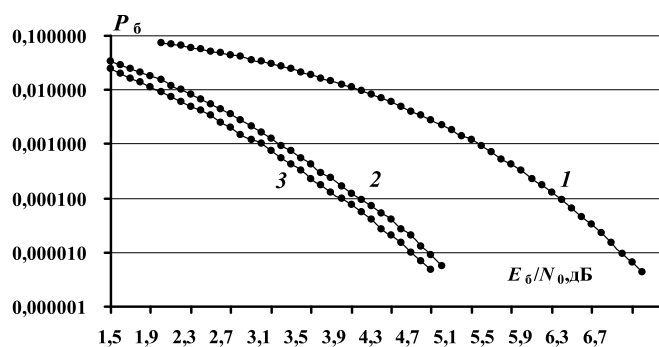


Рис. 2. Вероятности ошибки на бит итеративного приема турбокода, формируемого на основе низкоплотного кода (15,7) (длина кодовых слов  $n = 225$ , размерность  $k = 49$ , кодовая скорость  $R = 0,22$ ):

1 — 2 итерации; 2 — 5 итераций; 3 — 10 итераций

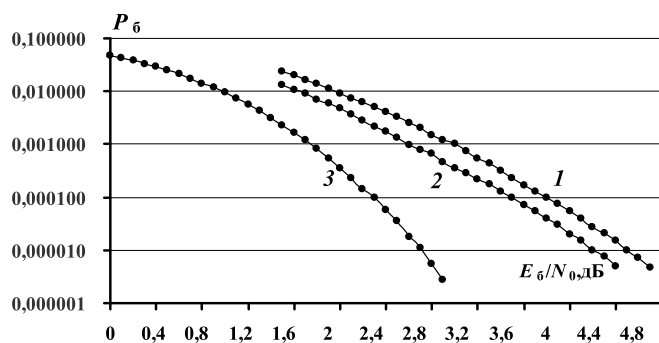


Рис. 3. Вероятности ошибки на бит итеративного приема турбокода, формируемого на основе низкоплотного кода (15,7) (длина кодовых слов  $n = 225$ , размерность  $k = 49$ , кодовая скорость  $R = 0,22$ , 10 итераций):

1 — алгоритм итеративного приема ВР; 2 — алгоритм итеративного приема  $m$ -ВР; 3 — граница вероятности  $P_6$  случайного кодирования

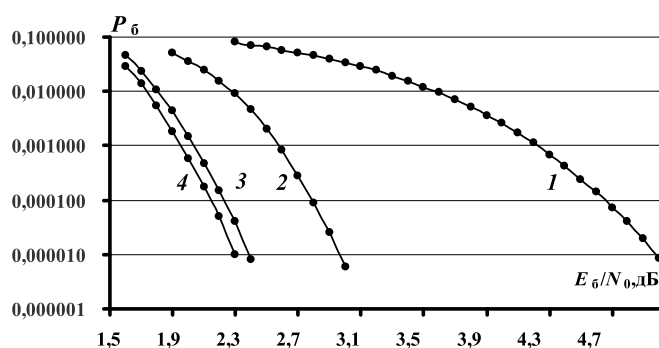


Рис. 4. Вероятности ошибки на бит итеративного приема турбокода, формируемого на основе низкоплотного кода (63,37) (длина кодовых слов  $n = 3969$ , размерность  $k = 1369$ , кодовая скорость  $R = 0,34$ ):

1 — 2 итерации; 2 — 5 итераций; 3 — 10 итераций; 4 — 15 итераций

на основе низкоплотных кодов, параметры которых приведены в табл. 2. Кривые соответствуют зависимостям вероятности ошибки на бит  $P_6$  от параметра сигнал/помеха при наличии АБГШ с односторонней спектральной плотностью  $N_0$ . Параметр сигнал/помеха определяется как  $E_6/N_0$ , где  $E_6$  — энергия сигналов на информационный бит. Число испытаний при моделировании ограничивалось достижением не менее 500 ошибок.

На рис. 2 приведены вероятности  $P_6$  для разного числа итераций алгоритма ВР для турбокода, формируемого на основе низкоплотного кода (15,7) (длина кодовых слов  $n = 225$ , размерность  $k = 49$ , кодовая скорость  $R = 0,22$ ). Кривые 1, 2 и 3 соответствуют применению двух, пяти и 10 итераций. Кривая 3 также соответствует применению 15 итераций, поэтому применение 10 итераций обеспечивает сходимость итеративного приема. При применении 10 итераций значение  $P_6 = 10^{-5}$  достигается при  $E_6/N_0 = 4,5$  дБ.

На рис. 3 приведены вероятности  $P_6$  при реализации 10 итераций итеративного приема рассматриваемого турбокода. Кривая 1 соответствует алгоритму ВР, кривая 2 — алгоритму итеративного приема  $m$ -ВР, кривая 3 — известной границе  $P_6$  случайного кодирования [1, 9] для данного турбокода, которая показывает существование кода с параметрами  $(n, k)$  и граничными вероятностными характеристиками. Видно, что применение алгоритма  $m$ -ВР определяет энергетический выигрыш до 0,2 дБ по отношению к алгоритму приема ВР. Видно также, что вероятностная кривая для алгоритма приема  $m$ -ВР близка к теоретической границе  $P_6$  случайного кодирования — для  $P_6 = 10^{-5}$  различие не превышает 1,5 дБ.

На рис. 4 приведены вероятности  $P_6$  для разного числа итераций алгоритма ВР для турбокода, формируемого на основе низкоплотного кода (63,37) (длина кодовых слов  $n = 3969$ , размерность  $k = 1369$ , кодовая скорость  $R = 0,34$ ). Кривые 1, 2, 3 и 4 соответствуют применению двух, пяти, 10 и 15 итераций. Видно, что кривые 3 и 4 практически совпадают (энергетические отличия не превышают 0,1 дБ), поэтому применение 10 итераций практически обеспечивает сходимость итеративного приема. При применении 10 итераций значение  $P_6 = 10^{-5}$  достигается при  $E_6/N_0 = 2,4$  дБ.

На рис. 5 приведены вероятности  $P_6$  при реализации 10 итераций итеративного приема рассматриваемого турбокода (см. рис. 4). Кривая 1 соответствует алгоритму ВР, кривая 2 соответствует алгоритму итеративного приема  $m$ -ВР. Кривая 3 соответствует границе  $P_6$  случайного кодирования для данного турбокода.

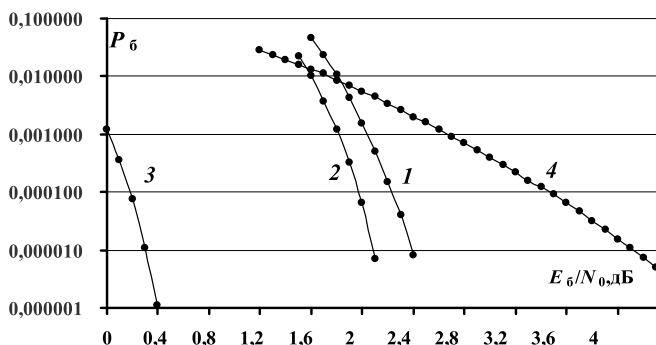


Рис. 5. Вероятности ошибки на бит итеративного приема турбокода, формируемого на основе низкоплотного кода (63,37) (длина кодовых слов  $n = 3969$ , размерность  $k = 1369$ , кодовая скорость  $R = 0,34$ , 10 итераций):

1 — алгоритм итеративного приема VR; 2 — алгоритм итеративного приема  $m$ -VR; 3 — граница вероятности  $P_6$  случайного кодирования; 4 — вероятность ошибки сверточного кода с кодовой скоростью  $1/3$  (длина кодового ограничения 7, алгоритм приема Витерби)

Видно, что применение алгоритма  $m$ -VR также определяет энергетический выигрыш до 0,2 дБ по отношению к алгоритму приема VR. Видно также, что вероятностная кривая для алгоритма приема  $m$ -VR близка к теоретической границе  $P_6$  случайного кодирования — для  $P_6 = 10^{-5}$  различие не превышает 1,8 дБ. Вероятностная кривая 4 соответствует сверточному коду с эквивалентной кодовой скоростью  $R = 1/3$  (длина кодового ограничения 7, алгоритм приема Витерби) [9]. Для  $P_6 = 10^{-5}$  энергетический выигрыш рассматриваемого турбокода с использованием алгоритма итеративного приема  $m$ -VR по отношению к сверточному коду составляет 2 дБ.

### Заключение

Объект настоящих исследований — блочные турбокоды, особенность которых заключается в том, что они формируются с использованием составляющих низкоплотных кодов и также являются низкоплотными кодами. Вследствие этого при приеме данных турбокодов можно применить алгоритмы итеративного приема, разработанные для общего класса низкоплотных кодов, в частности алгоритм VR и его модификацию  $m$ -VR.

Приведены результирующие характеристики рассматриваемых турбокодов (длительность кодовых слов, информационный объем, кодовая скорость, вероятностные характеристики при приеме), формируемых с использованием ряда составляющих низкоплотных кодов на основе конечной Евклидовой геометрии, которые входят в класс циклических кодов и характеризуются низкой сложностью алгоритмов формирования и итеративного приема.

Для ряда рассматриваемых турбокодов проведено моделирование алгоритмов приема VR и  $m$ -VR, при использовании которых не требуется знания относительно значений энергетического параметра канала сигнал/помеха. Путем моделирования показано, что алгоритм  $m$ -VR более эффективен по сравнению с алгоритмом VR (энергетический выигрыш достигает 0,2 дБ).

Путем компьютерного моделирования показано, что энергетический выигрыш для турбокода по отношению к сверточному коду с эквивалентными кодовыми скоростями  $R \approx 1/3$  для  $P_6 = 10^{-5}$  составляет 2 дБ.

Анализ характеристик турбокодов, формируемых с использованием класса составляющих низкоплотных кодов на основе конечной проективной геометрии в дополнение к конечной Евклидовой геометрии, представляет перспективное направление исследований.

Работа выполнена при поддержке РФФИ (№ 16-07-00746).

### Список литературы

1. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976. 594 с.
2. Pyndiah R. M. Near-optimum decoding of product-codes: block turbo-codes // IEEE Transactions on Communications. 1998. Vol. 46, N. 8. P. 1003—1010.
3. Головкин И. В., Назаров Л. Е. Разработка и реализация алгоритмов итеративного декодирования блочных турбокодов // Цифровая обработка сигналов. 2009. № 4. С. 37—40.
4. Назаров Л. Е., Батанов В. В., Кузнецов О. О. Алгоритмы итеративного посимвольного приема блочных турбокодов на основе кодов с проверкой на четность. // Журнал радиоэлектроники (электронный журнал). 2014. № 9. URL: <http://jre.cplire.ru/jre/sep14/1/text.pdf>.
5. MacKay D. J. C., Neal R. M. Near Shannon limit performance of low density parity check codes // Electronics Letters. 1997. Vol. 33. P. 457—458.
6. Johnson S. J. Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes. Cambridge: Univ. Press, 2010. 335 p.
7. Назаров Л. Е., Щеглов М. А. Характеристики полных и укороченных помехоустойчивых низкоплотных кодов на основе конечных геометрий // Успехи современной радиоэлектроники. 2017. № 6. С. 23—30.
8. Назаров Л. Е. Алгоритмы посимвольного приема сигналов // Информационные технологии. 2010. № 2. С. 53—55.
9. Зюко А. Г., Фалько А. И., Панфилов И. П., Банкет В. Л., Ивашенко П. В. Помехоустойчивость и эффективность систем передачи информации. М.: Радио и связь, 1985. 272 с.
10. Liu Z., Pados D. A. A decoding algorithms for finite-geometry LDPC codes. // IEEE Transactions on Communications. 2005. Vol. 53, N. 3. P. 415—421.
11. Tanner R. M. A recursive approach to low complexity codes // IEEE Transactions on Information Theory. 1981. Vol. 27, N. 9. P. 533—547.
12. Назаров Л. Е., Щеглов М. А. Вероятностные характеристики итеративного приема дискретных сигналов на основе помехоустойчивых блочных низкоплотных кодов // Журнал радиоэлектроники (электронный журнал). 2015. № 4. Режим доступа: <http://jre.cplire.ru/jre/apr15/5/text.pdf>.

L. E. Nazarov, D. Sc., Leading Researcher, e-mail: nazarov@ire.rssi.ru,  
Kotel'nikov Institute of Radio-Engineering and Electronics of RAS, Fryazino  
P. V. Shishkin, e-mail: p\_v\_s@bk.ru,  
JSC Academician M. F. Reshetnev "Information Satellite Systems", Zheleznogorsk

## The Characteristics of Error-Correcting Block Turbo-Codes Based on Low-Density Parity-Check Codes

The focus of this paper is directed towards the investigation of the characteristics of error-correcting block turbo-codes which enable communication at relatively low received signal/noise and provide very high power efficiency. Specific feature of investigated turbo-codes is construction with usage of low-density parity-check codes (LDPC) and these turbo-codes are in the class of LDPC too. According to this fact the considered turbo-codes have symbol-by-symbol decoding algorithms developed for total class LDPC codes, namely, BP (belief propagation) and BP modification (mBP) decoding algorithms. The BP and mBP decoding algorithms are iterative and for that implementation the signal/noise is not required. The resulted characteristics of turbo-codes constructed with usage of LDPC based on Euclidean geometry (namely, duration of code words, information volume, code rate, error performances) are presented in this paper. These component LDPC codes are cyclic and have encoding and decoding algorithms with low complexity implementation. The computer simulations for encoding and iterative decoding algorithms for the number of turbo-codes with different code rate and information volumes are performed. The results of computer simulations have shown that the mBP decoding algorithm is more effective than BP decoding algorithm (difference between signal/noise is more 0.2 dB). The simulations have shown that the error performances of investigated turbo-codes are better than that for convolutional codes with equivalent code rate 1/3, the degradation is about 2 dB for bit-error 0.00001.

**Keywords:** block product codes, low-density parity-check codes, iterative decoding

### References

1. Peterson W. W., Weldon E. J. *Error-Correcting Codes*. The MIT PRESS Cambridge, Massachusetts and London, England. 1972.
2. Pyndiah R. M. Near-optimum decoding of product-codes: block turbo-codes, *IEEE Transactions on Communications*, 1998, vol. 46, no. 8, pp. 1003–1010.
3. Golovkin I. V., Nazarov L. E. *Razrabotka i realizacija algoritmov iterativnogo dekodirovaniya blokovich turbo-kodov* [The development and realization of block turbo-codes decoding], *Zifrovaja obrabotka signalov*, 2009, no. 4, pp. 37–40.
4. Nazarov L. E., Batanov V. V., Kuznezov O. O. Algoritmi iterativnogo posimvolnogo priema blokovich turbo-kodov na osnove kodov s proverkoj na chetnost [Symbol-by-symbol iterative decoding algorithms for turbo-codes based on single-parity-check codes] // *Zhurnal Radioelektroniki*, 2014, no. 9, available at: <http://jre.cplire.ru/jre/sep14/1/text.pdf> (in Russian).
5. MacKay D. J. C., Neal R. M. Near Shannon limit performance of low density parity check codes, *Electronics Letters*, 1997, vol. 33, pp. 457–458.
6. Johnson S. J. *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*, Cambridge: Univ. Press, 2010.
7. Nazarov L. E., Sheglov M. A. Charakteristiki polnich I ukorochenich nizkoplotnostnich kodov na osnove ronechnich geometrii [The Characteristics of Full and Shortened Immune-Noise LDPC Codes Based on Finite-Geometry], *Uspechi sovremennoi radioelektroniki*, 2017, no. 6, pp. 23–30 (in Russian).
8. Nazarov L. E. Algoritmi posimvolnogo priema signalov [The algorithms of symbol-by-symbol decoding], *Informacionnyye tehnologii*, 2010, no. 2, pp. 53–55.
9. Zuko A. G., Falko A. I., Panfilov I. P., Banket V. L., Ivachenko P. V. *Pomechoustoichivost i effektivnost system peredachi informacii* [The noise-immunity and effectiveness of system message transmission]. Moscow, Radio i svyaz, 1985 (in Russian).
10. Liu Z., Pados D. A. A decoding algorithms for finite-geometry LDPC codes, *IEEE Transactions on Communications*, 2005, vol. 53, no. 3, pp. 415–421.
11. Tanner R. M. A recursive approach to low complexity codes, *IEEE Transactions on Information Theory*, 1981, vol. 27, no. 9, pp. 533–547.
12. Nazarov L. E., Sheglov M. A. Veroyatnostnie charakteristiki iterativnogo priema diskretnih signalov na osnove pomehoystoichivih nizkoplotnostnich kodov (The error-performances of symbol-by-symbol iterative decoding algorithms for signals based on low-density parity-check codes), *Zhurnal Radioelektroniki*, 2015, no. 4, available at: <http://jre.cplire.ru/jre/apr15/5/text.pdf>.

---

### Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: [it@novtex.ru](mailto:it@novtex.ru)

Технический редактор Е. В. Конова.

Корректор З. В. Наумова.

Сдано в набор 05.04.2018. Подписано в печать 24.05.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ618. 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.