DOI 10.17587/issn.1684-6400

ISSN 1684-6400

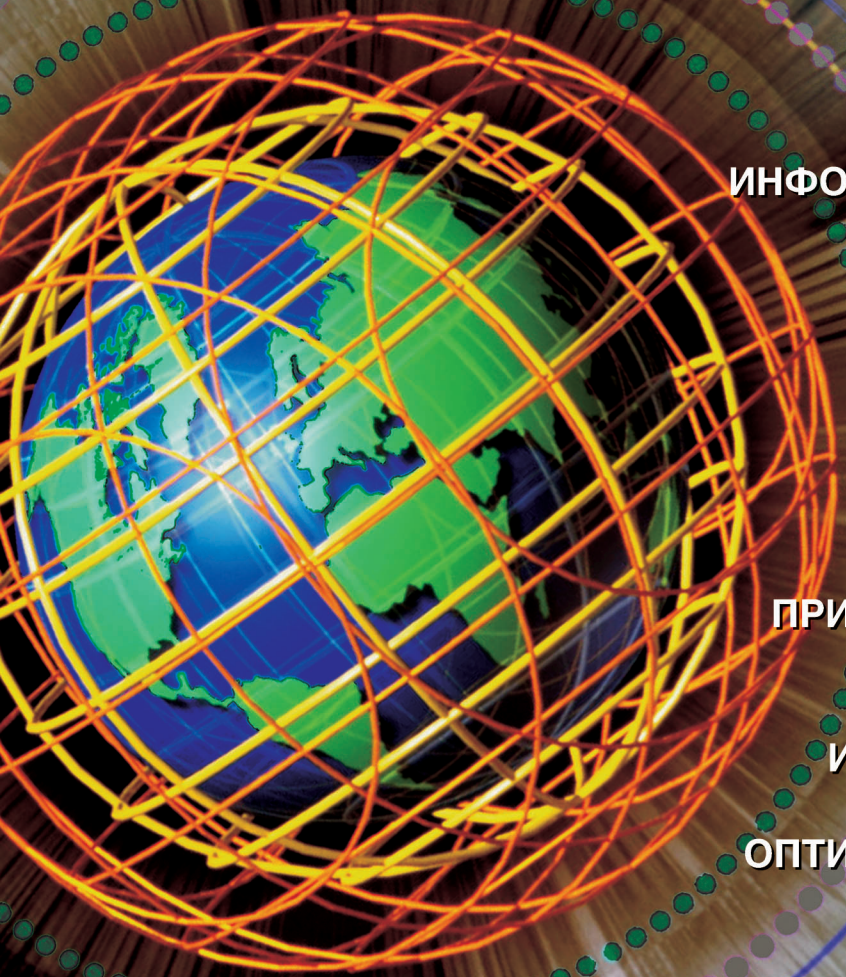
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 26

2020

№ 1

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ



САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

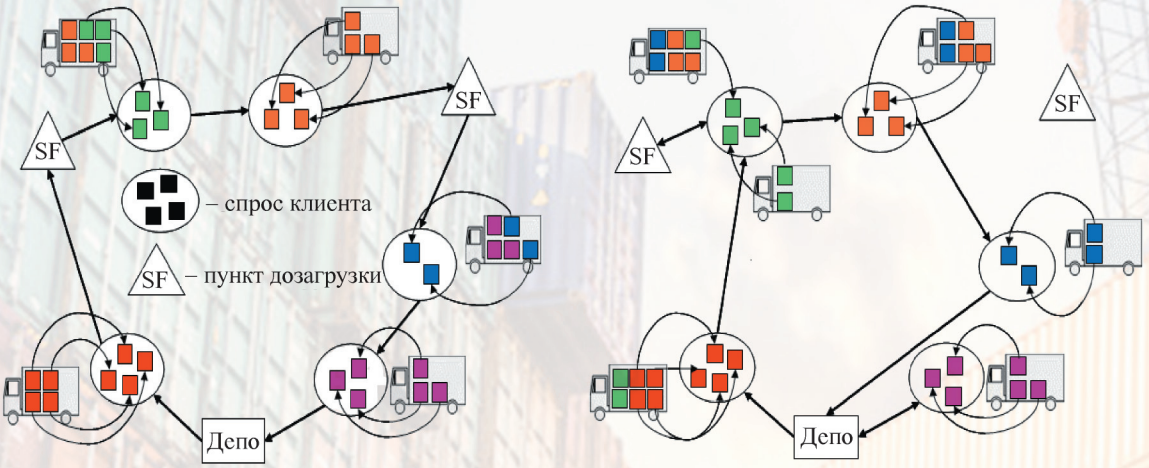
ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

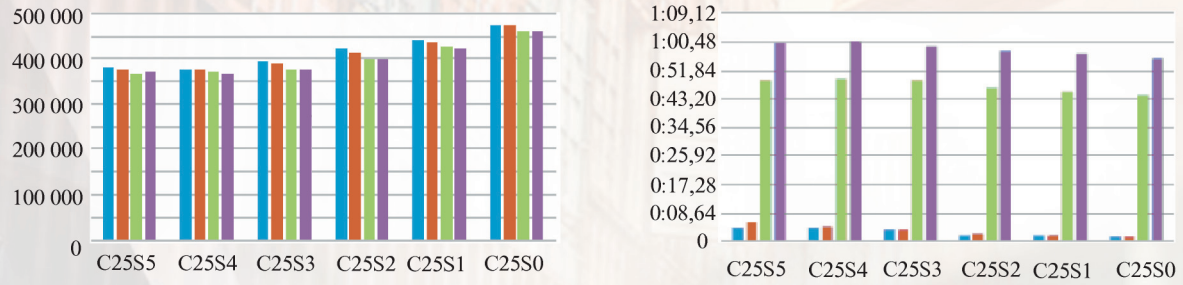
Рисунки к статье А. Ф. Валеевой, И. А. Янтурина, Р. С. Валеева

«ОБ ОДНОЙ ЗАДАЧЕ ДОСТАВКИ ГРУЗА РАЗЛИЧНЫМ КЛИЕНТАМ С ВОЗМОЖНОСТЬЮ ДОЗАГРУЗКИ»



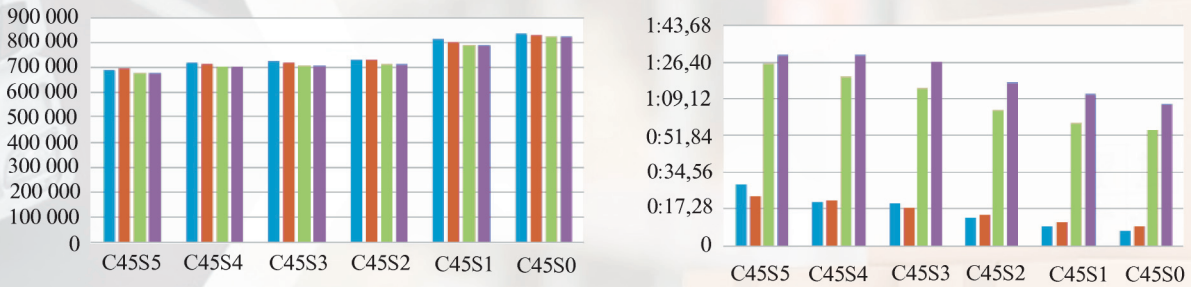
а) *б)*

Рис. 2. Иллюстрация доставки груза клиентам с раздельной доставкой и без нее:
а – маршрут доставки груза без раздельной доставки; *б* – маршрут доставки груза с раздельной доставкой



а) *б)*

Рис. 4. Сравнение средних стоимостей маршрутов (*а*) и среднего времени работы алгоритмов (*б*) для 25 клиентов: – (1+1)-EA(400I); – (1+1)-EA(500I); – (1+2)-EA(400I); – (1+2)-EA(500I)



а) *б)*

Рис. 5. Сравнение средних стоимостей маршрутов (*а*) и среднего времени работы алгоритмов (*б*) для 45 клиентов: – (1+1)-EA(400I); – (1+1)-EA(500I); – (1+2)-EA(400I); – (1+2)-EA(500I)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 26
2020
№ 1

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

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

INFORMATION TECHNOLOGIES
INFORMACIONNYE TEHNOLOGII

Vol. 26
2020
No. 1

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

Complete Internet version of the journal at site: **http://novtex.ru/IT.**
According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ
MODELING AND OPTIMIZATION

DOI: 10.17587/it.26.3-8

S. Khashin, PhD, Professor, e-mail: khash2@gmail.com,
S. Vaganov, PhD Student, e-mail: pro100-pioner@mail.ru,
Ivanovo State University, Ivanovo, 153025, Russian Federation,

Corresponding author: **Khashin Sergei,** Ivanovo State University,
Ivanovo, 153035, Russian Federation, e-mail: khash2@gmail.com

Genetic Algorithms Using Forth

A method for automatic finding of a program (in the FORT language) that realizes the given algorithm is developed. The algorithm is specified by a set of tests of the form (input_data) → (output_data). Both input and output data are represented as a sets of 4-byte integers. Genetic methods have made it possible to find the implementation of even relatively complex algorithms: decimal and binary digits of numbers, GCD, LCL, factorial, simple divisors, binomial coefficients, sorting of short sequences, highs, lows, calculation of polynomial values and others. The genetic approach allows you to build a program from separate blocks, "genes", which turned out to be suitable for at least some part of the test elements. Genetic methods made it possible to find the implementation of even relatively complex algorithms: decimal and binary digits of a number,, NOC, factorial, simple divisors, binomial coefficients, sorting of short sequences, maxima, minima, calculation of polynomial values and others. Our method starts by randomly sorting through short programs, extracting blocks ("genes") from them at least slightly suitable for the problem being solved. And then he builds a program using the found "genes". The set of used "genes" in the process of the algorithm is constantly being adjusted, improved. The complexity of direct enumeration grows exponentially with increasing program length. The genetic method we propose allows us in many cases to drastically reduce the volume of search. The FORT language is chosen because of its compactness: all listed algorithms are placed in no more than 10—15 commands. Although, if we take into account the genes found, the total length of the program will be significantly longer. In addition, the mechanism of embedding "genes" already, in fact, is in the language. Our method is configured to work with integers, but it can be applied to data containing real numbers, strings, etc. In the case of working with real numbers, the method can be considered as an alternative to the methods used in neural networks.

Keywords: *Genetic algorithm, Linear genetic programming, Evolutionary programming, Machine learning, Forth*

УДК 004.023

DOI: 10.17587/it.26.3-8

С. И. Хашин, PhD, канд. физ.-мат. наук, доц., e-mail: khash2@gmail.com,
С. Е. Ваганов, аспирант, e-mail: pro100-pioner@mail.ru,
Ивановский государственный университет, г. Иваново

Genetic Algorithms Using Forth

Разработан метод автоматического нахождения программы (на языке ФОРТ), реализующей данный алгоритм. Алгоритм задается в виде набора тестов (входные данные) → (выходные данные). И входные, и выходные данные представлены в виде наборов целых 4-байтовых чисел.

Генетический подход позволяет строить программу из отдельных блоков, "генов", которые оказались подходящими хотя бы для некоторой части тестовых элементов. Генетические методы позволили найти реализацию даже сравнительно сложных алгоритмов: десятичные и двоичные цифры числа, НОД, НОК, факториал, простые делители, биномиальные коэффициенты, сортировка коротких последовательностей, максимумы, минимумы, вычисление значений полиномов и другие.

Наш метод начинает работу со случайного перебора коротких программ, выделяет из них блоки ("гены") хоть немного подходящие для решаемой задачи. А затем строит программу с использованием найденных "генов".

Комплект используемых "генов" в процессе работы алгоритма постоянно корректируется, улучшается.

Сложность прямого перебора растет экспоненциально с ростом длины программы. Предлагаемый нами генетический метод позволяет во многих случаях радикально сократить объем перебора.

Язык ФОРТ выбран ввиду его компактности: все перечисленные алгоритмы помещаются в не более, чем 10—15 команд. Хотя, если учитывать найденные гены, общая длина программы будет существенно больше. Кроме того, механизм встраивания "генов" уже фактически есть в языке.

Наш метод настроен на работу с целыми числами, однако его можно применить и к данным, содержащим действительные числа, строки и т. д. В случае работы с действительными числами метод можно рассматривать как альтернативу методам, применяемым в нейронных сетях.

Ключевые слова: *генетический алгоритм, линейное генетическое программирование, эволюционное программирование, машинное обучение, Forth*

Inroduction

Our goal is to automatically obtain a program in some programming language that implements an algorithm, defined by a set of test's elements. To create such a program, we will use methods that in some sense have analogies in biology, in genetics. This approach is called "Genetic programming" (GP) [2, 6]. In all known applications, the desired algorithm is defined by a set of tests, for example, for the sum of two squares we would have:

$$1, 1 \to 2$$
$$2, -1 \to 5$$
$$0, 4 \to 16$$
$$\ldots$$

In the early days of GP, the goal of GP was exactly the construction of a program that implements a given algorithm in the selected programming language. In this paper we return to this original goal and show that it can work effectively. Moreover, in the classical textbooks [7, 8], it was emphasized that the particular choice of a programming language is not of importance, as all languages are equivalent in power to the Turing machine.

In practice, this was not entirely true. The number of possible programs turned out to be too large and was highly dependant on the choice of the programming language. So generally speaking, the application of GP for searching programs did not give the results that they had hoped for at the beginning.

Significant achievements were obtained in the case of tree-based genetic programming, that is, when the program is presented in the form of a tree of calculations. However, in this case the obtained algorithms are much more complicated than we wish them to be. The classical example which is frequently used in textbooks and articles considers a program that implements computation $x \to x^2 + x + 1$ or even $x \to x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ [9, 17, 20, 25]. These GP obtained programs have no cycles. In this direction, there is an interesting work [18]. Authors managed to find programs that implement the following functions:

- nguen1: $x \to x^3 + x^2 + x + 1$,
- nguen2: $x \to x^4 + x^3 + x^2 + x + 1$,
- nguen3: $x \to x^5 + x^4 + x^3 + x^2 + x + 1$,
- nguen4: $x \to x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$,
- keijzer4: $x \to x^3 e^{-x}\cos(x)(\sin^2(x)\cos(x) - 1)$,

and several other functions of this kind.

Genetic methods yielded great results in approximating real-valued functions. Here GP methods can compete with neural networks and even interlock with them. For example, in [26] typical 10 regression problems for neural networks are solved using GP methods.

In [12, 15], the concept of Recurrent Cartesian Genetic Programming (RCGP) is introduced, in which the calculation graph can contain cycles. As a result, the authors managed to implement the functions $n \to n(2n - 1)$, $n \to 1 + n(n + 1)/2$, $n \to n(n^2 + 1)/2$, and finally, even the Fibonacci numbers. However, this is not a real replacement for full-cycle operators.

In [13], 29 tasks were proposed for testing genetic programming research. Most of them require working with real numbers, strings, arrays, with displaying information on the screen. Some of these tasks can be also handled by our approach if we leave the result in the stack instead of its output onto the screen.

The approach closest to ours is Linear Genetic Programming (LGP) method [6, 11, 17, 20, 23, 24]. In this method, the program is represented as a sequence of instructions from imperative programming language. Their typical set consists of 4 arithmetic operations (two arguments) and a number of mathematical functions, for example exp, sin, sqrt, etc. Practically all considered systems receive algorithms without cycles.

In [14] C++ is used. However, here the primary goal is the (minor) correction of already created (by students) programs.

In [10] describes automatic creation of the simplest program in the language Haskell. Again there are no cycles in the examples, even though there are calls to functions that work with arrays as a whole.

In [21] several GP methods are benchmarked on the 29 tasks from [13].

In the book [16] methods of GP in Python are considered. Here, more than a dozen different problems are solved by GP methods. The main difficulty in them is not the genetic algorithms themselves, but the suitable reformulation of the problem. After that, the problem can be solved by the genetic method, or directly.

The widely distributed machine learning package "TensorFlow" [3] also contains some tools for genetic programming. See more on this in, e.g., [19].

In fact, in almost all cases, the considered genetic methods are reduced to finding the minimum of certain loss function from some, possibly, quite a considerable number of parameters.

All these approaches takes us far away from the original idea of GP. In the this paper we instead stay within the framework of the originally formulated task. Algorithms are assumed to be quite complex, including conditional operators and cycles. Obviously, for a linear representation of the program one needs some version of the bytecode. Another problem is the method of accessing variables. In our opinion, one of the most compact way is the data stack. Programming language Forth language satisfies these conditions. Functions in Forth take all the arguments from the data stack and also leave the results of calculations there. In this paper, we restrict ourselves to the minimum possible set of tools: we don't use real numbers, strings, only 4-byte integers, we do not use variables, arrays, only data in the stack. Thus, we reduce Forth to a minimal subset of tools (only 32-bit integers, only stack, no registers, no variables), which allow us to solve quite complicated tasks, like GCD, binomial coefficient, primality testing, and so on. In the future, we would like to introduce other data types: real numbers, strings, etc. In addition, from the rather exotic language Forth, we would like to switch to some more common language (C, Java, etc.) or to an assembly language.

The key idea is to use partial programs, that is, programs that do not pass all tests, but only a part of them. We assume that fragments of such programs will be contained in the general program. We will use these fragments as "genes".

Overfitting. When training neural networks, the problem of overfitting often arises. In our case, this means that the program will work correctly only on the input data that are available in the test. This means that the program implements a tabular algorithm. This problem is completely eliminated in the present work as we introduce a limit on the length of the program. It is sufficient to require that the length of the program does not exceed the number of test elements.

Structure of the work. In section 1, we define the concept of a test and a test element. Section 2 contains a brief description of the version of Forth that we use. In section 3 we describe the methods that are used to find the Forth-realization, section 4 describes our results.

1. Problem definition

We want to find a program in some programming language implementing given algorithm. The algorithm will be defined as a set of tests:

$$(input\_data) \to (output\_data).$$

The unit of the test is the pair $t = (x, y)$, where $(x \in X, y \in Y)$, and X, Y are the corresponding sets of input and output objects. In this paper we consider as input and output objects as a sequences of 32-bit integers of fixes length.

Test files. A "test" is an arbitrary finite set of test items. Sometimes, when there are no confusion, we call a single test element a test. In practice, the test is represented by a text file of the form:

```
#T SHIFTR _ 0 x,y -> x*2^y comment
<in> 9 3 </in><out> 1536 </out>
<in> 4 3 </in><out> 48 </out>
<in> 2 7 </in><out> 28 </out>
...
```

Definition 1. Pair (m, n), where m is a number of input integers and n is a number of output integers is called the signature of the test element.

Remark 1. In general, it is allowed to have test elements with different signatures in one test. But this possibility is not used in the present work.

The number of test items in each test varies from a several to hundreds.

A large set of test files is freely available on our website [1]. Not for all of them our GP method was able to find implementing programs. The tests on the website are grouped by sections: polynomials, max/min, sorting, GCD, LCM, factorials, number theory, decimal and binary digits and so on.

2. Programming language

From our point of view all programs are divided into two types: linear (with goto) and structural.

The advantages and disadvantages of each type are quite obvious and there is no fundamental difference between them from our point of view. We have chosen a linear approach in our work (at least for this initial stage). Usually, this is done by selecting an assembly-type instruction set (registers). In the present work the stack approach (already implemented in the language of Forth [4, 5]) is deemed to be more effective. To find the program with the genetic method, the most stripped-down version of the Forth was chosen. Only those instructions are left, which are impossible to do without. There is no interaction with the user, such as output to the screen, and even variables are missing. The control structures are represented only by unconditional and conditional jump-instruction. Forth is very compact, new words (functions) are introduced very easily. The newly defined functions have exactly the same syntax as the built-in language elements. This is convenient for a genetic approach. Thanks to this, one does not need to go through programs with complex structure in the form of a tree, only the simplest ones.

An example of a program that calculates the sum of the squares of two numbers and the factorial:

```
: SUMSQ2 DUP * SWAP DUP * +;
: FACTORIAL CONST 1 OVER -- -ROT *
OVER IF -6 SWAP DROP;
```

All programs and functions in Forth work with the data stack. Only 4-byte integers are stored on the stack. The functions have no explicit arguments. The input data that they take is from the stack and they leave the results of their work in the same place.

The number of such implicit arguments can vary, depending on the state of the stack. Such a signature will be called "floating". For now, we will only consider functions with a fixed, static signature. At each point of such a program, the current stack depth is statically determined.

Everywhere below, functions will be called "words", or "genes". At each moment, the system has a certain set of built-in words (functions) and the current set of new genes, that is, new words built in the learning process ("evolution"). One step of evolution is the construction of one or more new words (functions) that solve one problem (test) in whole or in part.

Bytecode. The program in Forth is a byte sequence, each byte is a separate command. There are two types of commands: built-in and implemented on Forth itself. Thus, the total number of commands cannot be more than 255.

In our version of Forth there are 33 built-in commands: unconditional (GOTO) and conditional (IF) jump, numeric literal (CONST), stack manipulation commands (DUP DROP SWAP OVER ROT — ROT 2PICK 3PICK 4PICK 3ROLL 4ROLL), arithmetic (NEGATE +− * / % /% ++ − −), bit commands (AND OR XOR NOT), logical (comparison) (> < = 0 = 0> 0>).

In addition to the built-in commands, the system may contain commands implemented on Forth itself, for example, finding the sum of squares of two elements at the stack:

```
: SUMSQ2        DUP * SWAP DUP * +;
```

The colon at the beginning of the command is a sign of the beginning of a new word. The name of the function follows. The semicolon at the end is a sign of the end of the word (function, program). The name of a function can be an arbitrary string of characters that does not contain spaces, tabs to the end of the line. Individual words are separated by spaces or tabs or line ends. An important advantage of Forth is that it is not required to develop a mechanism for "embedding genes", as it is already exist in the language. New words (procedures, functions, "genes") are used in exactly the same way as built-in ones. The current list of words (the dictionary) under biological interpretation will be considered as a "genome".

3. General algorithm

At first, let's just try to iterate through all the programs to a given length.

Working time. Having 33 basic words, we get $\approx 10^9$ programs of length 6.

If we check about 10 million programs per second (one core, frequency ≈3 GHz), then a full search will take about a minute. In an hour, you can go through all the programs of length 7. The search of all programs of length 8 requires more than a day. For an 8-core processor, it is realistic to check programs of length 9. Finding longer programs requires more complex methods.

A base step with parameters (L_0, L_1, T) in a fixed dictionary is the following algorithm.

1. A full search of programs up to length $\leqslant L_0$. Then build a list of partial programs (that is, programs that do not perform all tests, but only some of them).

2. Using the partial programs, build a list of word frequencies and a list of frequencies of word pairs.

3. Check random programs of length $L_0 + 1 ... L_1$ within a specified interval of time T seconds.

4. Correct the frequency tables and repeat step 3, this cycle is performed K (usually 8) times.

The typical values of the parameters for working on one processor core is (7, 14, 400), that is, first perform a full search of programs of length $\leqslant 7$, then 8 times for 50 seconds is performed alternately probabilistic/Markov search of programs of length from 8 to 14 with correction of the frequency table after each cycle.

Genes. As already mentioned, the list of words in the dictionary plays the role of "genome". Each gene added can be either "good", "useful", or "unsuccessful", "useless". The quality of the function-gene itself cannot be determined, only as part of the genome, aimed at solving a specific problem. Therefore, we introduce a definition of genome (dictionary) quality in relation to this set of tests. If the new gene improves the quality, we will consider it a good one. As the "candidates for genes" we will take the most frequent chains of bytes of length 2 or 3 among partial programs. During the process we will have two lists of programs (genes): "candidates for genes", and "unsuccessful genes". There should be no duplicate elements in them.

1. Find a list of valid words. Upon completion of the search, we get, besides this list, also a list of partial programs.
2. Clear the list "candidates for the genes" and "bad genes".
3. In the lists of "candidates for genes" add the most frequent chains of length 2, 3, and only those that are not in the list of "unsuccessful genes".
4. If there are no candidates for genes, we finish the job.
5. Add one of the candidates to the dictionary.
6. Perform a basic step with this genome, find its quality.
7. If the gene was unsuccessful, remove it from the dictionary and add it to the list of "unsuccessful genes".
8. Goto step 3.

4. Results

Here is a list of algorithms that have been implemented.

By brute force and probabilistic method. Squaring, odd, even, abs, sign, sum of two squares, max/min of two or three numbers, sorting of two numbers, GCD in special case ($x,y \geqslant 1$), factorial ($x \geqslant 1$), left/right shift in $k>0$ bits, minimal natural divider in special case ($x>1$). Decimal/hex/binary digits: lower, high.

```
Squaring::          SQUARE DUP *;
Multiply by 2:      MUL2 DUP +;
Test odd:     ODD CONST 1 AND;
Signum:       SIGN DUP 0 = IF 6 0> DUP IF 2 --;
Sum of suares: SUMSQ2 DUP * SWAP DUP * +;
Absolute value: ABS DUP 0> IF 2 NEG;
Descending sort of 2 numbers:
              SORT2R OVER OVER > IF 2 SWAP;
Ascending sort of 2 numbers:
              SORT2 DUP 2PICK > IF 2 SWAP;
Maximum of 2 numbers:
              MAX2 OVER OVER > IF 2 SWAP DROP;
Minimum of 2 numbers:
              MIN2 DUP 0> IF 2 SWAP IF 2 ROT;
Maximum of 3 numbers:
              MAX3 OVER 3ROLL > IF 2 SWAP DROP;
Minimum of 3 numbers:
              MIN3 ROT 2PICK 0> IF -4 DROP DROP;
Polynomials:
(a,b,x) → bx + a:     poly1 * +;
(a,b,c,x) → cx^2 + bx + a:
              poly2 -ROT 2PICK * + * +;
x → 2x-3:     pol1 _ 1 -- DUP + --;
x → -3x + 4:  pol1 _ 2 -- CONST -3 * + +;
x → x-1:      pol1 _ 3 --;
x → 2x-1:     pol1 _ 4 DUP + --;
x→ 10x-3:     pol1 _ 5 CONST 10 * -- -- --;
x -> 11x + 7: pol1 _ 6 CONST 11 * CONST 7 +;
x → 2x^2:     pol2 _ 1 DUP OVER + *;
x → -3x^2 + x: pol2 _ 2 DUP CONST -3 * + + *;
x -> x^2 -2x + 1: pol2 _ 3 -- DUP *;
(a,b,c) → b^2-4ac: discr OVER ROT *
              -ROT * CONST 4 * -;
Digits:
Lower digit:  digit0 CONST 10 %;
Next digit:   digit1 CONST 10 / CONST 10 %;
Hundreds:     digit2 CONST 10 DUP * / CONST 10 %;
The highest digit: digitH DUP C10 / DUP -ROT
              IF -5 SWAP DROP;
```

Using genes. Using the constructed algorithms as "genes", we managed to find the following algorithms.

Sum of three squares, sorting/mid of three numbers, GCD in general case (arbitrary x,y), factorial (arbitrary x), left/right shift, minimal natural divider in general case (arbitrary x). Arbitrary decimal/hex/binary digits. The average of the three numbers. Binomial coefficients.

Here are, for example, some of these programs.

Right shift: (n,x) → 2^n x (only for $n>0$)

```
:     DUP + SWAP -- DUP -ROT IF -7 +;
```

Digit K: (K, x) → $(x/10^K)$ % 10

```
:     digitK DUP IF 4 SWAP C10 / OVER --
      ROT IF -7 DROP C10 %;
```

GCD(x,y):

```
:     GCD _ 0 DUP -ROT % DUP IF -5 -;
:     GCD _ 1 OVER ROT IF 3 GOTO -5 GCD _ 0;
:     GCD ABS SWAP ABS GCD _ 1;
```

or, without genes:

```
:      GCD ABS SWAP ABS OVER ROT
       IF 3 GOTO -5 DUP -ROT % DUP
       IF -5 -;
:      FACTORIAL CONST 1 OVER -- -ROT *
       OVER IF -6 SWAP DROP;
:      BINOM DUP 2PICK – SWAP FACTORIAL
       SWAP FACTORIAL / SWAP FACTORIAL /;
```

Programs are given in full, with already built-in genes.

An exe-file that implements the algorithms described in the work can be obtained on our website [1]. A detailed description of the program is available in our text in arxiv.org [26] (this text contains a very detailed description of our programs which we removed from the present paper for the purpose of conciseness).

Conclusion

The proposed GP method allows to obtain programs that implement quite complex algorithms: factorial, binomial coefficients, search for a prime divider, etc. The achieved level of program complexity significantly exceeds that obtained by other methods. However, there are still many algorithms that cannot be implemented. For example, the finding of Fibonacci numbers is not yet obtained without "prompts", that is, without additional genes.

Based on the analysis of the obtained results, it is possible to build more advanced methods that can enable the implementation of even more complex algorithms.

References

1. **Gen_03.exe,** available at: http://math.ivanovo.ac.ru/dalgebra/Khashin/gene/gen_03r.htm (date:04.06.2018).

2. **genetic-programming.com**, available at: http://www.genetic-programming.com/ (date:04.06.2018).

3. **TensorFlow**: An open source machine learning framework for everyone, available at: https://www.tensorflow.org/ (date:04.06.2018).

4. **Thinking Forth Project,** available at: http://thinking-forth.sourceforge.net/ (date:30.06.2018).

5. **Wikipedia: Forth,** available at: URL:https://en.wikipedia.org/wiki/Forth_programming_language \\(date:30.06.2018).

6. **Wikipedia: LGP** Linear genetic programming, available at: https://en.wikipedia.org/wiki/Linear_genetic_programming \\ (date:04.06.2018).

7. **Koza, J. R.** Genetic programming as a means for programming computers by natural selection, *Stat Comput.*, 1994, vol. 4(2), pp. 87—112, doi:10.1007/BF00175355.

8. **Banzhaf W., Nordin P., Keller R. E., Francone F.** Genetic Programming — An Introduction; *On the Automatic Evolution of Computer Programs and its Applications*, M., Kaufmann Publ. Inc. CA, USA, 1998, 480 p.

9. **Panchenko T. V.** Genetic algorithms, Astrakhan Univ. publ., 2007, 87 p.

10. **Katayama S.** MagicHaskeller on the Web: Automated Programming as a Service, *In Haskell '13: Proceedings of the 2013 ACM SIGPLAN Symposium on Haskell. ACM.*

11. **Francoso L., Sotto D. P., de Melo V. V.** Comparison of linear genetic programming variants for symbolic regression, *GECCO'14. Proc. 2014 Annual Conference on Genetic and Evolutionary Computation*, doi: 10.1145/2598394.2598472.

12. **Turner A. J., Miller J. F.** Recurrent Cartesian genetic programming applied to famous mathematical sequences, *Proceedings of the 7th York Doctoral ymposium on Computer Science&Electronics*, 2014, cs.york.ac.uk, pp. 37—47.

13. **Helmuth T., Spector L.** General Program Synthesis Benchmark Suite, *GECCO'15. Proc. 2015 Annual Conference on Genetic and Evolutionary Computation*, p. 1039—1046.

14. **Yalin K., Stolee Kathryn T., Le C., Brun Y.** Repairing Programs with Semantic Code Search., *ASE'15, Proceedings of the 2015 30th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*, pp. 295—306, doi: 10.1109/ASE.2015.60.

15. **Turner A. J., Miller J. F.** Neutral genetic drift: an investigation using Cartesian Genetic Programming, *Genet Program Evolvable Mach* (2015), doi:10.1007/s10710-015-9244-6.

16. **Sheppard C.** Genetic Algorithms with Python, 2016, 282 p.

17. **Chen Q., Xue B., Shang L., Zhang M.** Generalisation of Genetic Programming for Symbolic Regression with Structural Risk Minimisation, *GECCO '16 Proceedings of the Genetic and Evolutionary Computation Conference 2016* P. 709—716, doi:10.1145/2908812. 2908842.

18. **Sotto L. F. D. P., Melo V. V.** A Probabilistic Linear Genetic Programming with Stochastic Context-Free Grammar for solving Symbolic Regression problems, *arXiv:1704.00828* [cs.NE].

19. **Staats K., Pantridge E., Cavaglia M., Milovanov I., Aniyan A.** TensorFlow Enabled Genetic Programming, *arXiv:1708.03157* [cs.DC]

Simson J. Open-Source Linear Genetic Programming, *Ph.D. thesis, Waikato*, New Zealand.

21. **Pantridge Helmuth T., McPhee N., Spector L.** On the Diffculty of Benchmarking Inductive Program Synthesis Methods, *In Proceedings of GECCO '17 Companion*, Berlin, Germany, July 15—19, 2017, 8 p., doi:10.1145/3067695.3082533.

22. **Thi T. P., Nguyen X. H., Nguyen T. T.** A Study on Fitness Representation in Genetic Programming, *CTA 2016. Advances in Intelligent Systems and Computing*, vol 538, Springer, Cham, doi:10.1007/978-3-319-49073-113.

23. **Milano N., Nolfi S.** Scaling Up Cartesian Genetic Programming through Preferential Selection of Larger Solutions, *arXiv:1810.09485* [cs.NE].

24. **Wilson D. G., Miller J. F., Cussat-Blanc S., Luga H.** Positional Cartesian Genetic Programming, *arXiv:1810.09485* [cs.NE].

25. **Virgolin M., Alderliesten T., Witteveen C., Bosman P.** A Model-based Genetic Programming Approach for Symbolic Regression of Small Expressions, *arXiv:1904.02050* [cs.NE] (submitted for peer review to IEEE Transactions on Evolutionary Computation).

26. **Khashin S. I., Vaganov S. E.**, Genetic algorithms in Forth, *arXiv: 1807.06230* [math.NT].

УДК 519.7 DOI: 10.17587/it.26.9-15

А. Ф. Валеева, д-р техн. наук, проф., aida_val2004@mail.ru,
И. А. Янтурин, магистрант, yanturin.ilmir@gmail.com,
Р. С. Валеев, канд. техн. наук, доц., ruslan_valeev@inbox.ru,
Уфимский государственный авиационный технический университет

Об одной задаче доставки груза различным клиентам с возможностью дозагрузки

Рассматривается задача маршрутизации для доставки груза различным клиентам с учетом возможности дозагрузки недостающего заказа в соответствующих пунктах (Modified Vehicle Routing Problem with Satellite Facilities, MVRPSF). Представлена математическая модель задачи MVRPSF, включающая такие ограничения, как грузоподъемность транспортных средств (ТС), наличие депо и пунктов дозагрузки, раздельную доставку. Для ее решения разработан эволюционный алгоритм, позволяющий получать рациональные маршруты доставки груза различным клиентам с предварительным размещением заказов в ТС.

Ключевые слова: *маршрутизация, размещение грузов в автомобильные транспортные средства, пункты дозагрузки, эволюционный алгоритм*

Введение

При принятии логистических решений в целях удовлетворения требований клиентов по доставке груза (заказов) главными являются задачи операционного планирования [1], таких как: прогнозирование объемов спроса; управление запасами; управление поставками и закупками; транспортировка, включающая выбор типа транспортных средств (ТС), определение их маршрутов и плана загрузки; организация погрузочно-загрузочных работ; складирование. В данной статье основное внимание уделяется задачам маршрутизации с возможностью дозагрузки недостающих заказов в специализированных пунктах дозагрузки и их размещения в автомобильных ТС одинаковой грузоподъемности. Данные задачи представляются актуальными, поскольку транспортно-логистические услуги составляют 86 % от всего объема рынка российской логистики [2], а по данным 2018 г. доля автомобильных перевозок в России составляет 40,6 % [3].

Задачи маршрутизации ТС (*Vehicle Routing Problem*, *VRP*) и задачи размещения грузов внутри ТС (*Three-Dimensional Bin Packing Problem*, *3DBPP*), как известно, являются NP-трудными задачами комбинаторной оптимизации, и для их решения разрабатываются эвристические методы. Как отмечено в работе [3], применение современного логистического инструментария позволит снизить общие экономические издержки в среднем на 15...35 %, а транспортных расходы — примерно на 25 %.

Для поиска оптимальных маршрутов по классификации, предложенной в работе [4] P. Toth и D. Vigo, известны следующие задачи маршрутизации ТС: с ограничением на грузоподъемность (*Capacitated Vehicle Routing Problem, CVRP*), когда каждое ТС имеет ограниченную грузоподъемность; с временными окнами (*Vehicle Routing Problem with Time Windows, VRPTW*), когда у каждого клиента есть так называемое "временное окно", в которое он должен быть обслужен; с множеством депо *(Multiple Depot Vehicle Routing Problem, MDVRP*), если используются несколько депо для обслуживания клиентов; с раздельной доставкой (*Split Delivery Vehicle Routing Problem, SDVRP*), когда каждый клиент может быть обслужен одновременно несколькими ТС; с заданным периодом планирования в размере нескольких дней (временной горизонт), если клиенты посещаются с разной частотой, заданной для каждого клиента (*Periodic Vehicle Routing Problem, PVRP*); с немедленным возвратом товаров (*Vehicle Routing Problem with Pick-up and Delivery, VRPPD*), когда клиенты могут возвращать некоторые товары в депо; с возвратом товаров (*Vehicle Routing Problem with Backhauls, VRPB*), если клиенты могут возвращать некоторые товары в депо, но только после того, как весь товар будет доставлен клиентам; с возможностью дозагрузки (*Vehicle Routing Problem with Satellite Facilities, VRPSF*), когда ТС могут дополнительно загрузиться на маршруте в промежуточных пунктах-складах; со случайными данными (*Stochastic Vehicle Routing Problem, SVRP*), когда некоторые ком-

поненты задачи (число и спрос клиентов, расстояния между городами и клиентами) могут иметь случайное поведение.

В статье рассматривается задача маршрутизации *MVRPSF*, являющаяся модификацией известной задачи *VRPSF* [5], возникшая в компании, занимающейся организацией освещения в различных регионах России. Одной из подзадач являлась доставка соответствующего оборудования (шкафов *АСКУЭ* (*АСКУЭ* — автоматизированная система коммерческого учета энергоресурсов (рис. 1)) арендуемыми автомобильными ТС одинаковой грузоподъемности, находящимися в депо. При этом имеются пункты дозагрузки, в которых ТС могут пополнять недостающий запас груза последующим клиентам, не возвращаясь в депо. Известны спрос на оборудование, время обслуживания клиентов (населенные пункты), максимально разрешенное время маршрута, стоимость пройденной единицы времени ТС (включены стоимости аренды ТС и бензина), стоимость прохождения ТС по платным дорогам, дорожные ограничения. Груз (оборудование) доставляется клиентам в контейнерах параллелепипедной формы, предварительно размещенных на поддонах. Требуется найти маршрут доставки оборудования с минимальными общими затратами при выполнении следующих ограничений:

- каждый маршрут начинается и заканчивается в депо;
- масса груза, загружаемого в ТС, не должна превышать грузоподъемности ТС (*CVRP*);
- каждый клиент может быть обслужен более чем одним ТС (запрос на поставку груза клиенту может быть разделен между несколькими ТС — используется раздельная доставка, *SDVRP*);
- перед отправкой к различным клиентам груз размещается в ТС (решается задача *3DBPP*);
- ТС могут выполнить дозагрузку на маршруте, без возврата в депо с помощью дополнительных пунктов дозагрузки.

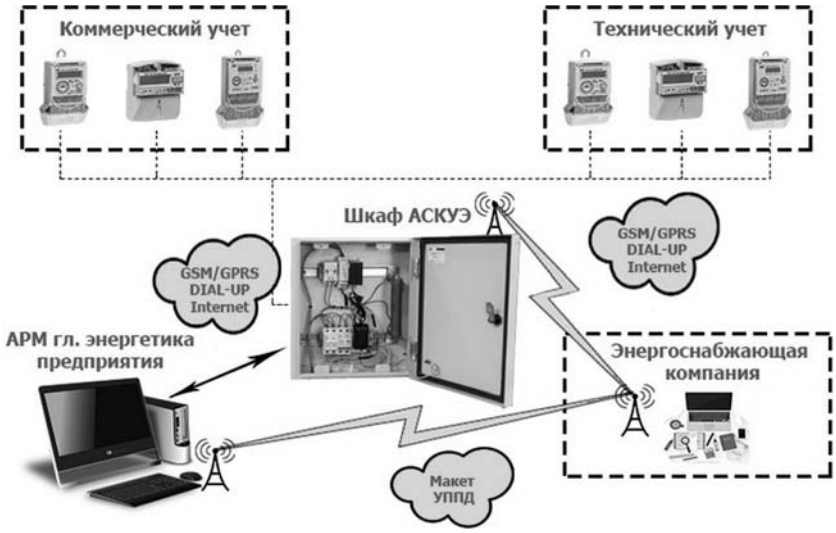


Рис. 1. Структурная схема *АСКУЭ*

1. Математическая модель задачи доставки груза различным клиентам с возможностью дозагрузки (*MVRPSF*) и задачи его размещения внутри ТС (*3DBPP*)

В основу модели задачи доставки груза различным клиентам с возможностью дозагрузки (*MVRPSF*) с учетом ряда ограничений легли модели следующих задач: задача маршрутизации ТС с ограничением на грузоподъемность (*CVRP*), задача маршрутизации ТС с раздельной доставкой (*SDVRP*), задача маршрутизации ТС с возможностью дозагрузки (*VRPSF*) [5], причем обозначения, введенные в перечисленных задачах, сохраняются. На рис. 2, *а*, *б* (см. вторую сторону обложки) приведена иллюстрация доставки груза клиентам с раздельной доставкой и без, причем груз, предназначенный различным клиентам, окрашен в разные цвета.

Математическая модель задачи MVRPSF

Дано: $G = (V, E)$ — взвешенный граф с множеством вершин V и множеством ребер E (под ребром понимаются участки автомобильных дорог); V — множество вершин, которое делится на два подмножества I и F; I — множество клиентов; $I_0 = I \cup \{0\}$, где вершина 0 — это депо; i, j — номера клиентов; $i, j \in I = \{1, ..., n\}$; F — множество пунктов дозагрузки; α — индекс для пунктов дозагрузки или депо, $\alpha \in F = \{1, ..., s\} \cup \{0\}$; F_μ — множество вершин, соответствующее потенциальным посещениям пунктов дозагрузки; F_0 — депо, используемое в качестве пункта для дозагрузки; n_α — число дополнительных пунктов для дозагрузки; $\overline{n} = n + \sum_{\alpha=0}^{s} n_\alpha$ — число пунктов в сети; d_{ij} — расстояние между пунктами i и j, где пунктами могут быть клиенты,

депо или дозагрузочные пункты; τ_{ij} — время в пути между пунктами i и j; m — число используемых ТС; Q_v — грузоподъемность ТС v; $\hat{Q}_v$ — максимальное количество разрешенного в ТС v остатка груза до его пополнения, $\hat{Q}_v < Q_v$; $\cos_{rent}$ — стоимость аренды ТС в единицу времени; $\cos t_{petrol}$ — затраты на бензин ТС; $\cos t_{road_{ij}}$ — стоимость прохождения по платным дорогам между клиентами i, j; $i, j \in I = \{1, ..., n\}$; T_v — максимально разрешенное время маршрута для ТС v; q_i — спрос клиента i; p_{iv} — время обслуживания клиента i ТС v, время загрузки в дозагрузочных пунктах, либо в депо;

Переменные: x_{ij}^v — переменная логического типа, принимающая значение 1, если ТС v перемещается из депо в направлении от клиента i к клиенту j, и 0 — в противном случае; t_j^v — время начала обслуживания клиента j; y_{jv} — загрузка остатка груза (оборудования) непосредственно перед посещением клиента j; $\hat{y}_{iv}$ — количество груза, доставляемого клиенту i ТС v.

Требуется минимизировать общие транспортные расходы:

$$\sum_{\substack{i,j=0 \\ i \neq j}}^{\bar{n}} \sum_{v=1}^{m} (\cos t_{rent} + \cos t_{petrol} + \cos t_{road_{ij}}) d_{ij} x_{ij}^v \to \min \quad (1)$$

при следующих ограничениях:

- пройденный маршрут включает каждую вершину не менее одного раза (*SDVRP*):

$$\sum_{\substack{i,j=0 \\ i \neq j}}^{\bar{n}} \sum_{v=1}^{m} x_{ij}^v \geqslant 1, \; i = 1, \ldots, n; \quad (2)$$

- каждый дозагрузочный пункт имеет максимум одного преемника (клиента, к которому ТС v едет после дозагрузки):

$$\sum_{\substack{j=1 \\ i \neq j}}^{n} x_{ij}^v \leqslant 1, \; i \in F_0 \cup \ldots \cup F_s, v = 1, \ldots, m; \quad (3)$$

- число посещений клиентов равно числу выездов от них:

$$\sum_{\substack{i=0 \\ i \neq j}}^{\bar{n}} x_{ji}^v - \sum_{\substack{i=0 \\ i \neq j}}^{\bar{n}} x_{ij}^v = 0, \; j = 0, \ldots, \bar{n}, \; v = 1, \ldots, m; \quad (4)$$

- количество груза y_{jv} в ТС v перед посещением клиента j:

$$\begin{aligned} & y_{jv} \leqslant \bar{y}_{iv} x_{ij}^v + \bar{Q}_{iv}(1 - x_{ij}^v), i \neq j; \\ & i \in I \cup F_0 \cup \ldots \cup F_s; \\ & j \in I_0 \cup F_0 \cup \ldots \cup F_s; v = 1, \ldots, m; \end{aligned} \quad (5)$$

- количество груза в ТС v, если j — клиент:

$$\hat{y}_{jv} \leqslant y_{jv} \leqslant Q_v, j = 1, \ldots, n; \; v = 1, \ldots, m; \quad (6)$$

- количество груза в ТС v, если j — дозагрузочный пункт:

$$0 \leqslant y_{jv} \leqslant \hat{Q}_v, i \in F_0 \cup \ldots \cup F_s; v = 1, \ldots, m; \quad (7)$$

- клиент i может быть обслужен ТС v в том случае, если ТС v проходит через i (*CVRP*):

$$\bar{y}_{iv} \leqslant q_i \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij}^v \leqslant 1, i = 1, \ldots, n; \; v = 1, \ldots, m; \quad (8)$$

- спрос всех клиентов в грузе должен быть удовлетворен (*CVRP*):

$$\sum_{v=1}^{m} \hat{y}_{iv} = q_i, \; i = 1, \ldots, n; \quad (9)$$

$$x_{ij}^v \in \{0,1\}; \; i = 0, \ldots, n; \; j = 0, \ldots, n; \; v = 1, \ldots, m;$$

где

$$\bar{y}_{iv} = \begin{cases} y_{iv} - \hat{y}_{iv}, \text{ если } i \in I; \\ Q_v, \text{ если } i \in F_0 \cup \ldots \cup F_s; v = 1, \ldots, m; \end{cases}$$

$$\bar{Q}_{iv} = \begin{cases} Q_v - \hat{y}_{iv}, \text{ если } i \in I; \\ Q_v, \text{ если } i \in F_0 \cup \ldots \cup F_s; v = 1, \ldots, m. \end{cases}$$

Перед доставкой груза клиентам он размещается в кузове автомобильного ТС (решается задача упаковки *3DBPP*), загрузка проводится от задней части ТС (от дверей) последовательно в порядке обхождения клиентов по правилу "*LIFO* (*Last-In First-Out*)" с учетом центра тяжести.

Математическая модель задачи 3DBPP

Дано:

W — ширина кузова ТС; L — длина кузова ТС; H — высота кузова ТС; $pod = (x_{pod}, y_{pod}, z_{pod})$ — координаты переднего левого нижнего угла поддона;

(l_{pod}, w_{pod}, h_{pod}, q_{pod}, $h_{pod\ \max}$) — размеры поддона, где l_{pod} — длина; w_{pod} — ширина; h_{pod} — высота; q_{pod} — грузоподъемность; $h_{pod\ \max}$ — максимально разрешенная высота поддона; ($x_{pod\ jl}$, $y_{pod\ jl}$, $z_{pod_{jl}}$) — координаты l-й вершины j-го поддона, j = 1, ..., *kolpod*, где *kolpod* — число поддонов; N — число заданных контейнеров с грузом для различных клиентов; w = = (w_1, ..., w_i, ..., w_N) — вектор ширин контейнеров; l = (l_1, ..., l_i, ..., l_N) — вектор длин контейнеров; h = (h_1,..,h_i,...,h_N) — вектор высот контейнеров; m = (m_1, m_2, ..., m_N) — вектор масс контейнеров, где m_i — масса i-го контейнера, C = (cg_1, cg_2, ..., cg_N) — вектор центров тяжести

контейнеров, где $cg_i = (x_{i\text{ц.т}}, y_{i\text{ц.т}}, z_{i\text{ц.т}})$ — вектор координат центра тяжести i-го контейнера по осям x, y и z.

Вводится система координат XYZ, где (0,0,0) — начало координат, совпадающее с передней левой нижней вершиной груза (параллелепипеда), (x_i, y_i, z_i) — координаты i-го груза. Решение задачи представляется в виде: $<X, Y, Z, CG>$, $X, Y, Z, CG \in Z+$, где $X = (x_1, x_2, ..., x_i, ..., x_N)$, $Y = (y_1, y_2, ..., y_i, ..., y_N)$, $Z = (z_1, z_2, ..., z_i, ..., z_N)$ — векторы координат переднего левого нижнего угла размещенных грузов; $CG = (X_{\text{ц.т}}, Y_{\text{ц.т}}, Z_{\text{ц.т}})$ — координаты расположения центра тяжести загруженного кузова ТС допустимой области G, т. е. в цилиндре, R — радиус основания цилиндра, h_1, h_2 — границы значения допустимой высоты цилиндра:

$$G = (x - L/2)^2 + (y - W/2)^2 = R^2;$$
$$h_1 \leqslant z \leqslant h_2;$$
$$X_{\text{ц.т}} = \sum_{i=1}^{N} (x_i m_i) \Big/ \sum_{i=1}^{N} m_i;$$
$$Y_{\text{ц.т}} = \sum_{i=1}^{N} (y_i m_i) \Big/ \sum_{i=1}^{N} m_i;$$
$$Z_{\text{ц.т}} = \sum_{i=1}^{N} (z_i m_i) \Big/ \sum_{i=1}^{N} m_i,$$

где x_i, y_i, z_i — координаты центра тяжести i-го груза, m_i — масса i-го груза.

Требуется минимизировать число ТС с размещенными в них грузами:

$$NN \underset{P}{\rightarrow} \min,$$

где P — множество различных размещений грузов, при заданных ограничениях:

- предметы не выходят за грани поддона:

$$(x_i \geqslant 0) \wedge (y_i \geqslant 0) \wedge (z_i \geqslant 0) \wedge (l_{pod} > (x_i + l_i)) \wedge \\ \wedge (w_{pod} > (y_i + w_i)) \wedge (h_{pod} > (z_i + h_i)); \quad (10)$$

- стороны предметов параллельны граням поддона:

$$((x_{jl} = x_j) \vee (x_{jl} = (x_j + l_{pod})) \wedge \\ \wedge ((y_{jl} = y_j) \vee (y_{jl} = (y_j + w_{pod})) \wedge \\ \wedge ((z_{jl} = z_j) \vee (z_{jl} = (z_j + h_{pod})); \quad (11)$$

- стороны поддонов параллельны граням кузова ТС:

$$((x_{pod_{jl}} = x_{pod_j}) \vee (x_{pod_{jl}} = (x_{pod_j} + l_{pod_j})) \wedge \\ \wedge ((y_{pod_{jl}} = y_{pod_j}) \vee (y_{pod_{jl}} = (y_{pod_j} + w_{pod_j})) \wedge \\ \wedge ((z_{pod_{jl}} = z_{pod_j}) \vee (z_{pod_{jl}} = (z_{pod_j} + h_{pod_j})); \quad (12)$$

- поддоны, принадлежащие одному ТС, не перекрывают друг с другом:

$$(x_{pod_i} \geqslant (x_{pod_j} + l_{pod_j}) \vee \\ \vee x_{podj} \geqslant (x_{pod_i} + l_{pod_i})) \vee \\ \vee (y_{pod_i} \geqslant (y_{pod_j} + w_{pod_j}) \vee \\ \vee (y_{pod_j} \geqslant (y_{pod_i} + w_{pod_i})) \vee \\ \vee (z_{pod_i} \geqslant (z_{pod_j} + h_{pod_j}) \vee \\ \vee (z_{pod_j} \geqslant (z_{pod_i} + h_{pod_i})); \quad (13)$$

- поддоны не выходят за грани кузова:

$$(x_{pod_i} \geqslant 0) \wedge (y_{pod_i} \geqslant 0) \wedge (z_{pod_i} \geqslant 0) \wedge \\ \wedge (L \geqslant (x_{pod_i} + l_{pod_i})) \wedge (W \geqslant y_{pod_i} + w_{pod_i})) \wedge \quad (14) \\ \wedge (H \geqslant (z_{pod_i} + h_{pod_i}));$$

- максимальная высота $H_{\max}$ размещенных на поддон предметов не превышает максимально разрешенной высоте поддона:

$$H_{\max} \leqslant h_{pod\,\max}; \quad (15)$$

- стороны предметов (грузов) параллельны граням кузова ТС:

$$((x_{jl} = x_j) \vee (x_{jl} = x_j + l_j)) \wedge ((y_{jl} = y_j) \vee \\ \vee (y_{jl} = y_j + w_j)) \wedge ((z_{jl} = z_j) \vee (z_{jl} = z_j + h_j)),$$

где (x_{jl}, y_{jl}, z_{jl}) — координаты l-й вершины j-го груза для $j = 1, ..., N$;

- предметы не перекрывают друг друга:

$$(x_i \geqslant (x_j + l_j) \vee x_j \geqslant (x_i + l_i)) \vee (y_i \geqslant (y_j + w_j) \vee \\ \vee y_j \geqslant (y_i + w_i)) \vee (z_i \geqslant (z_j + h_j) \vee z_j \geqslant (z_i + h_i)), \\ \text{для } i \neq j,\ i, j = 1, ..., N; \quad (16)$$

- предметы не выходят за грани кузова ТС:

$$(x_i \geqslant 0) \wedge (y_i \geqslant 0) \wedge (z_i \geqslant 0) \wedge (L \geqslant (x_i + l_i)) \wedge \\ \wedge (W \geqslant (y_i + w_i)) \wedge (H \geqslant (z_i + h_i)); \quad (17)$$

- масса размещенных предметов на поддоне не превышает грузоподъемности поддона:

$$\sum_{i=1}^{N} m_i \leqslant q_{pod}; \quad (18)$$

- отклонение центра тяжести упакованного ТС от центра (точки пересечения диагоналей) должно быть минимальным:

$$d = \sqrt{(X_{\text{ц.т}} - L/2)^2 + (Y_{\text{ц.т}} - W/2)^2 + \\ + (Z_{\text{ц.т}} - H/2)^2} \rightarrow \min; \quad (19)$$

- координаты центра тяжести загруженного кузова ТС расположены в допустимой области G, т. е. в цилиндре:

$$(x - X_{\text{ц.т}})^2 + (y - Y_{\text{ц.т}})^2 \leqslant R^2;$$
$$h_1 \leqslant Z_{\text{ц.т}} \leqslant h_2. \qquad (20)$$

2. Эволюционный алгоритм для решения задач *MVRPSF* и *3DBPP*

Для решения рассматриваемой задачи разработан алгоритм *VRPSF-3D* на базе эволюционного алгоритма (μ + λ)-*EA* (*Evolutionary Algorithm, EA*) [6, 7]. В отличие от генетических алгоритмов, в алгоритме (μ + λ)-*EA* применяется только оператор мутации (случайным образом выбираются два клиента и меняются местами в маршруте), при этом параметр λ — это число потомков (новых маршрутов), а параметр μ — число родителей (сгенерированных маршрутов), при этом детерминировано выбираются лучшие по целевой функции потомки ($0 < \lambda < \mu$). На рис. 3 приведен алгоритм *VRPSF-3D* для решения задачи *MVRPSF*.

3. Численный эксперимент для оценки эффективности алгоритма *VRPSF-3D*

Численные эксперименты проводили на вычислительной машине Intel Core i5 с частотой 2,6 GHz и оперативной памятью 4 Гб на платформе 64-разрядной операционной системы Windows 8,1. Для решения задачи *MVRPSF* были реализованы два эволюционных алгоритма (1 + 1)-*EA* и (1 + 2)-*EA* и сгенерированы тестовые наборы (расстояния между клиентами, между клиентами и пунктами дозагрузки), значения спроса соответственно для 25 и 45 клиентов, пункты дозагрузки варьировались от 1 до 5. В тестовом наборе данных CnSn_αDq обозначение Cn — это число клиентов (n = 25; 45); Sn_α — число пунктов дозагрузки (n_α = 0, ..., 5); Dq — число прогонов (q = 7); 400I и 500I — число итераций (400 и 500) алгоритмов. Каждый тестовый набор прогонялся семь раз. В табл. 1 приведено сравнение средних стоимостей маршрутов для 25 клиентов и пяти дозагрузочных пунктов, на рис. 4, *а*, *б* (см. вторую сторону обложки) приведены диаграммы для результатов табл. 1 и среднего времени работы алгоритмов (1 + 1)-*EA* и (1 + 2)-*EA*.

В табл. 2 приведено сравнение средних стоимостей маршрутов для 45 клиентов и пяти дозагрузочных пунктов, на рис. 5, *а, б* (см. вторую сторону обложки) приведены диаграммы

Алгоритм VRPSF-3D

Шаг 0 . Инициализация переменных

Шаг 1. Загрузка груза в ТС v (решается *3DBPP* (10)-(20), [8])

Шаг 2. Цикл по числу итераций

Если маршрута нет, *то* генерировать λ маршрутов x_{total_q}

иначе

Цикл по λ

$x_q^{\lambda} = \text{мутация}(x_{total_q})$

Конец цикла по λ

Конец_ если_маршрута_нет

Цикл по числу маршрутов

$K = Q_v$ (решается *3DBPP*)

Цикл по числу клиентов n

Пока спрос i −го клиента q_i не будет удовлетворен, *выполнить*:

Если в ТС v все грузы i −го клиента размещены, (т.е. спрос $q_i \leq K$), *то* $K = K - q_i$; спрос i-го клиента удовлетворен

иначе

Если $q_i > Q_v$ *и если* доля оставшегося груза в ТС $K / Q_v \cdot 100\% \geq Q_{percent}$, *то*

$q_i = q_i - K$

Поиск ближайшего пункта дозагрузки s от i −го клиента

иначе

Поиск наименьшего расстояния от i −го клиента до пункта дозагрузки s и от пункта дозагрузки s до $i+1$−го клиента

Конец Если

Если найденный пункт дозагрузки s депо *и если* есть неиспользованные ТС, *то* новое ТС включается в маршрут

Добавление найденного пункта дозагрузки s в маршрут

$K = Q_v$ (решается *3DBPP*)

Конец_если $q_i \leq K$

Конец_цикла_пока

Конец цикла по числу клиентов

Конец_цикла_по_числу_маршрутов

Если выполняются ограничения (2)-(9), *то* сравнение стоимости нового и текущего маршрутов и выбор наилучшего маршрута, *иначе* маршрут не существует

Конец_цикла_по_итерациям

Вывод маршрута x_{total_q}

Рис. 3. Алгоритм *VRPSF-3D* для решения задачи *MVRPSF*

Таблица 1

Сравнение средних стоимостей маршрутов для 25 клиентов

Набор данных	Алгоритм			
	(1 + 1)-*EA*(400I)	(1 + 1)-*EA*(500I)	(1 + 2)-*EA*(400I)	(1 + 2)-*EA*(500I)
C25S5	379511,688	373939,880	365331,214	366879,060
C25S4	377455,978	377646,850	366516,772	365728,245
C25S3	389652,783	389228,222	373778,422	375115,648
C25S2	417374,221	410995,140	400887,474	398608,087
C25S1	436645,971	434722,761	425910,131	420026,363
C25S0	472241,377	468573,985	455591,205	456230,756

Таблица 2

Сравнение средних стоимостей маршрутов для 45 клиентов

Набор данных	Алгоритм			
	(1 + 1)-EA(400I)	(1 + 1)-EA(500I)	(1 + 2)-EA(400I)	(1 + 2)-EA(500I)
C45S5	691982,023	692911,677	675945,203	679537,390
C45S4	716029,800	714792,839	703425,290	699705,519
C45S3	724870,755	716403,365	707629,317	704713,844
C45S2	733388,352	728944,069	715523,360	710660,777
C45S1	811113,063	800260,047	789023,352	788447,802
C45S0	836895,690	832073,433	826121,102	822050,662

для результатов табл. 2 и среднего времени работы алгоритмов (1 + 1)-*EA* и (1 + 2)-*EA*.

Как показали численные эксперименты, лучший результат показал алгоритм (1 + 2)-*EA*, меньше вычислительного времени затрачивает алгоритм (1 + 1)-*EA*, при этом на стоимость маршрута оказывает влияние число пунктов дозагрузки: она меньше при их числе, равном пяти (в данном эксперименте).

Рассмотрим пример поиска маршрута для доставки оборудования в различные города Башкирии на реальных данных предприятия. Депо расположено в Уфе, клиенты находятся в шести городах: в Октябрьском (спрос $q_1 = 15$ на шкафы АСКУЭ), Стерлитамаке (спрос $q_2 = 25$), Ишимбае (спрос $q_3 = 9$), Кумертау (спрос $q_4 = 8$), Агидели (спрос $q_5 = 6$), Учалах (спрос $q_6 = 10$). Пункт дозагрузки в Салавате. Грузоподъемность ТС — 20 шкафов. Стоимость часа аренды ТС — 1000 руб. Расход ТС на 100 км равен 15 л (стоимость 1 л топлива равна 41,64 руб.), средняя скорость ТС — 90 км/ч. Алгоритмом (1 + 1)-*EA* найден маршрут Уфа — Стерлитамак — Салават — Стерлитамак — Ишимбай — Салават — Кумертау — Агидель — Нефтекамск — Октябрьский — Уфа — Учалы — Уфа, длина маршрута составила 1985 км, затраты (топливо и аренда ТС) равны 49 054 руб. (рис. 6). Алгоритмом (1 + 2)-*EA* получен маршрут Уфа — Стерлитамак — Салават — Стерлитамак — Ишимбай — Салават — Кумертау — Учалы — Уфа — Октябрьский — Нефтекамск — Агидель — Уфа, длина маршрута — 1791 км, затраты составили (на топливо и аренду ТС) 45 687 руб (рис. 7).

Стоимость маршрута состояла из стоимостей аренды ТС ($\cos t_{rent}$) и стоимости бензина ТС ($\cos t_{petrol}$), каждая из которых вычислялась соответственно по следующим формулам:

$$\cos t_{rent} = \left(\sum_{i=0}^{\bar{n}} \sum_{v=1}^{m} p_{iv} + \frac{\left(\sum\limits_{\substack{i,j=0 \\ i \neq j}}^{\bar{n}} \sum_{v=1}^{m} d_{ij}^{v} x_{ij}^{v} \right)}{средняя\_скорость\_TC\_v} \right) \times$$

$$\times стоимость\_аренды\_TC\_за\_ед.\_времени;$$

$$\cos t_{petrol} = \frac{\left(\sum\limits_{\substack{i,j=0 \\ i \neq j}}^{\bar{n}} \sum_{v=1}^{m} d_{ij}^{v} x_{ij}^{v} \right)}{100 \cdot расход\_топлива\_на\_100\ км} \times$$

$$\times стоимость\_одного\_литра\_топлива,$$

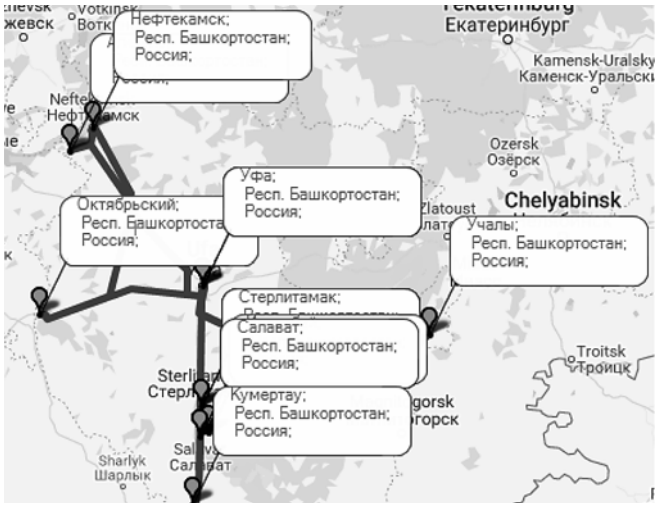


Рис. 6. Маршрут Уфа — Стерлитамак — Салават —Стерлитамак — Ишимбай — Салават — Кумертау — Агидель — Нефтекамск — Октябрьский — Уфа — Учалы — Уфа (алгоритм (1 + 1)-*EA*)

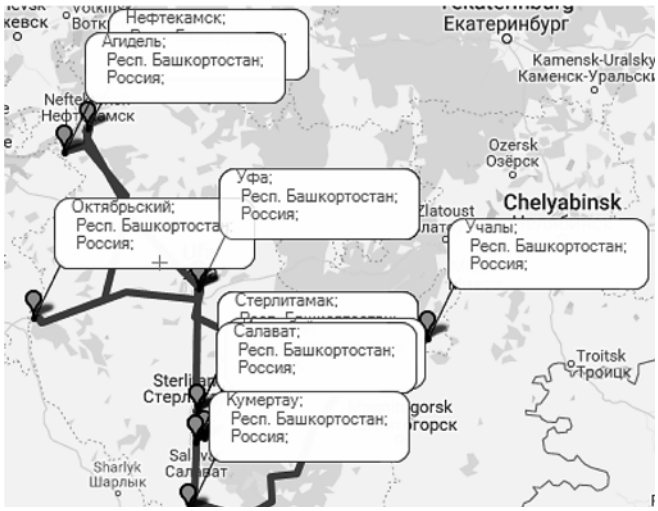


Рис. 7. Маршрут Уфа — Стерлитамак — Салават — Стерлитамак — Ишимбай — Салават — Кумертау — Учалы — Уфа — Октябрьский — Нефтекамск — Агидель — Уфа (алгоритм (1 + 2)-*EA*)

при этом средняя скорость ТС v равнялась 80 км/ч; расход топлива на 100 км — 15 л; стоимость одного литра топлива составила 45 руб., стоимость аренды ТС в ед. времени — 1000 руб.

Заключение

В статье были рассмотрены задачи операционного планирования, а именно: задача маршрутизации для доставки груза различным клиентам с учетом возможности дозагрузки и задача размещения его в ТС перед отправкой клиентам. Для задачи маршрутизации с возможностью дозагрузки предложена модификация известной математической модели, учитывающая при необходимости раздельную доставку груза (в том случае, когда груз клиента превышал грузоподъемность ТС).

Для перечисленных задач разработаны эволюционные алгоритмы (1 + 1)-*EA* и (1 + 2)-*EA*, учитывающие возможность дозагрузки и раздельную доставку груза, а также программное обеспечение, которое было апробировано на предприятии для доставки специального оборудования в различные города Башкирии. Поскольку рассматриваемая задача является новой, не существует как набора тестовых данных, так и результатов других методов решения для проведения сравнительного анализа. В связи с этим было сгенерировано 30 тестовых наборов данных. Был проведен численный эксперимент (в статье приведены лишь частичные результаты эксперимента), который показал, что решения, полученные с помощью эволюционного алгоритма (1 + 2)-*EA*, имеют лучшие значения целевой функции, чем решения, полученные с помощью (1 + 1)-*EA*, но алгоритм (1 + 1)-*EA* работает быстрее примерно в 3 раза.

Список литературы

1. **Langevin A., Riopel D.** Logistics Systems: Design and Optimization. New York: Springer, 2005. 388 p.
2. **Транспортная** стратегия Российской Федерации на период до 2030 г. URL: http://www.mintrans.ru/documents/detail.php?ELEMENT_ID.
3. **Моргунов В. И., Джабраилов А. Э.** Маркетинг. Логистика. Транспортно-складские логистические комплексы. М.: Издательско-торговая корп. "Дашков и К", 2010. 388 с.
4. **Toth P., Vigo D.** The Vehicle Routing Problem. Philadelphia: Society for Industrial and Applied Mathematics, 2002. P. 386.
5. **Bard J. F., Huang L., Dror M., Jaillet P.** A branch and cut algorithm for the VRP with satellite facilities // IIE Transactions (Institute of Industrial Engineers). 1998. Vol. 30, N. 9. P. 821—834.
6. **Rechenberg I.** Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen. Stuttgart: Fromman-Holzbook Verlag, 1973. 170 p.
7. **Борисовский П. А., Еремеев А. В.** О сравнении некоторых эволюционных алгоритмов //Автоматика и телемеханика. 2004. № 2. С. 3—9.
8. **Юсупова Н. И., Валеев Р. С.** Рациональное размещение грузов в контейнеры с учетом их физических характеристик с помощью роботизированного комплекса // Информационные технологии. 2011. № 5. С. 30—36.

A. F. Valeeva, D.Sc. Professor, aida_val2004@mail.ru, **I. A. Yanturin,** Master student, yanturin.ilmir@gmail.com,
R. S. Valeev, PhD., Associate Professor, ruslan_valeev@inbox.ru,
Ufa State Aviation Technical University, Ufa, 450008, Russian Federation

About One Problem of the Goods Delivery to Various Customers with Possibility of Additional Loading

The routing problem of the goods delivery for various customers accounting a possibility of additional loading for a missing order in correspondent points is considered in the paper (Modified Vehicle Routing Problem with Satellite Facilities, MVRPSF). The mathematical model of the MVRPSF problem is presented, it includes such restrictions as the carrying capacity of transport vehicles (TV), the presence of depots and additional loading points, the separate delivery. The evolutionary algorithm that provides obtaining the rational routes of the goods delivery to various customers with preliminary disposition of orders in a transport vehicle is developed.

Keywords: *routing; goods disposition in car transportation vehicles; additional loading points; evolutionary algorithms*

DOI: 10.17587/it.26.9-15

References

1. **Langevin A., Riopel D.** Logistics Systems: Design and Optimization, New York, Springer, 2005, 388 p.
2. **Transport** strategy of the Russian Federation for the period until 2030, available at: http://www.mintrans.ru/documents/detail.php?ELEMENT_ID (in Russian).
3. **Morgunov V. I., Dzhabrailov A. E.** Marketing. Logistics. Transport and warehouse logistics complexes, Moscow, Publishing and Trading Corp. "Dashkov and K", 2010, 388 p. (in Russian).
4. **Toth P., Vigo D.** The Vehicle Routing Problem. Philadelphia, Society for Industrial and Applied Mathematics, 2002, 386 p.
5. **Bard J. F., Huang L., Dror M., Jaillet P.** A branch and cut algorithm for the VRP with satellite facilities, *IIE Transactions (Institute of Industrial Engineers),* 1998, vol. 30, no. 9, pp. 821—834.
6. **Rechenberg I.** Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen, Stuttgart, Fromman-Holzbook Verlag, 1973, 170 p.
7. **Borisovsky P. A.,Eremeev A. V.** O sravnenii nekotorih evolucionnih algoritmov, *Avtomatica i Telemehanica*, 2004, no. 2, pp. 3—9 (in Russian).
8. **Yusupova N. I., Valeev R. S.** *Informacionnie Tehnologii*, 2011, no. 5, pp. 30—36 (in Russian).

УДК 621.391 DOI: 10.17587/it.26.16-21

С. В. Дворников, д-р техн. наук, проф., e-mail: practicdsv@yndex.ru,
А. В. Пшеничников, д-р техн. наук, доц., e-mail: siracooz77@mail.ru,
С. С. Манаенко, канд. техн. наук, ст. преп., e-mail: manaenkoss@mail.ru,
Военная академия связи им. С. М. Буденного

Статистическая модель помехозащищенных радиотехнических систем на основе порогового метода управления частотным ресурсом

Разработана статистическая модель помехозащищенных радиотехнических систем с пороговым методом управления частотным ресурсом. Сформулирована и доказана гипотеза о зависимости функции плотности распределения превышения уровня сигнала над уровнем помех в помехозащищенных радиолиниях от параметров массива исключаемых рабочих частот. Сделаны выводы по области практической реализации разработанных решений.

Ключевые слова: *помехозащищенные радиолинии, пороговый метод управления, частотный ресурс, уровень сигнала и помех, статистическая модель*

Введение

Актуальной проблемой разработки современных радиотехнических систем (РТС) является обеспечение требуемой устойчивости их функционирования в различных условиях сигнальной и помеховой обстановки. Существенные проблемы при решении задач данной предметной области возникают в условиях преднамеренных деструктивных воздействий (ПДВ).

Анализ реализуемых на практике режимов функционирования современных РТС показывает, что в условиях ПДВ широкое применение получил режим *FHSS (Frequency Hopping Spread Spectrum*) [1—5]. Вместе с тем в работах [3, 6, 7] показано, что реализация указанного режима *FHSS* основана на снижении эффективности функционирования РТС вследствие существенного увеличения требуемых ресурсов, а также существенной нагрузки на подсистему управления. При этом наиболее сложно рассмотренные задачи формализуются при представлении каналов радиосвязи моделями с неоднородными параметрами ввиду отсутствия соответствующего научно-методического аппарата.

Таким образом, возникает противоречие между реализацией режимов функционирования РТС в условиях неоднородных воздействий и практическими требованиями к их эффективности. С учетом указанных обстоятельств теоретические аспекты решения формализованных классов задач представлены в работе [8], в которой отражены результаты оценки статистических характеристик помехозащищенных радиолиний с управлением частотными ресурсами. Данная работа является логическим продолжением работы [8] и посвящена исследованию РТС с *FHSS* на основе порогового метода управления частотным ресурсом.

1. Формализация метода повышения эффективности РТС, использующих режим *FHSS*

Оригинальные подходы к решению задач синтеза методов повышения эффективности РТС отражены в достаточно большом числе исследований в данной предметной области. Наиболее распространенными из них являются подходы, основанные на разработке агрегированных сигнальных конструкций с повышенными свойствами помехоустойчивости при заданных значениях пропускной способности радиоканалов [9, 10].

Альтернативным подходом является использование методов, основанных на разработке сложных моделей РТС, учитывающих параметры потоков деструктивных воздействий различного характера [10, 11]. Особенностью данных решений являются возможности по технической реализации элементов РТС, учитывающих параметрические особенности ПДВ.

Вместе с тем общим недостатком представленных методов является низкая эффективность функционирования РТС в условиях наличия пораженных частотных каналов, используемых режимом *FHSS*, а также сложность в получении их количественных оценок при неоднородных параметрах радиоканалов.

Для решения такого класса задач воспользуемся подходами, предложенными в работах [10, 11]. Их особенность состоит в том, что они основаны на управлении ресурсами РТС, в частности частотно-временными. Так, в работе [12] показано, что такие алгоритмы обеспечивают наибольшую эффективность РТС в условиях ПДВ, о чем свидетельствует их практическая реализация, например в системах стандарта *ALE* 2*G* [13]. Учитывая указанные обстоятельства, обобщим данные подходы

с позиций решений, реализующих алгоритмы активного зондирования частотных каналов [14, 15], с последующей отбраковкой непригодных каналов, определив такой метод как пороговый метод управления частотным ресурсом. Указанные способы формализуем с позиций управления частотно-временным ресурсом РТС, использующих режим *FHSS* в условиях ПДВ.

Для дальнейших исследований представим радиоканал моделью канала прерывистой связи, учитывающей быстрые и медленные замирания, а также неоднородные условия сигнальной и помеховой обстановки на рабочих частотах.

Основываясь на известных результатах, приведенных, в частности, в [1], представим функции распределения огибающих сигналов $U_{\text{с}}$ и помех $U_{\text{п}}$ на рабочих частотах на основе закона Релея

$$W(U_{\text{с}}) = \frac{2U_{\text{с}}}{U^2_{\text{с эфф}}} \exp\left(-\frac{U^2_{\text{с}}}{U^2_{\text{с эфф}}}\right); \quad W(U_{\text{п}}) = \frac{2U_{\text{п}}}{U^2_{\text{п эфф}}} \exp\left(-\frac{U^2_{\text{п}}}{U^2_{\text{п эфф}}}\right) \tag{1}$$

и Райса

$$W(U_{\text{с}}) = \frac{2U_{\text{с}}}{U^2_{\text{с эфф}}} \exp\left(-\frac{U^2_{\text{с}} + U^2_{\text{с ср}}}{U^2_{\text{с эфф}}}\right) I_0\left(\frac{2U_{\text{с}}U_{\text{с ср}}}{U^2_{\text{с эфф}}}\right); \quad W(U_{\text{п}}) = \frac{2U_{\text{п}}}{U^2_{\text{п эфф}}} \exp\left(-\frac{U^2_{\text{п}} + U^2_{\text{п ср}}}{U^2_{\text{п эфф}}}\right) I_0\left(\frac{2U_{\text{п}}U_{\text{п ср}}}{U^2_{\text{п эфф}}}\right). \tag{2}$$

В выражениях (1) и (2) $U_{\text{с эфф}}$, $U_{\text{п эфф}}$ — эффективные напряжения флюктуирующей составляющей сигнала и помехи соответственно, являющиеся параметрами распределений; $U_{\text{с ср}}$, $U_{\text{п ср}}$ — амплитуды регулярной составляющей сигнала и помехи; I_0 — функция Бесселя нулевого порядка.

В работе [14] обосновано, что такое представление распределения огибающих сигналов и помех справедливо на коротких временных интервалах, измеряемых единицами минут, на протяжении которых параметры распределения $U_{\text{с эфф}}$ ($U_{\text{п эфф}}$) являются постоянными.

На более длительных временных интервалах параметры распределений сигналов и помех уже приобретают случайный характер и описываются плотностями вероятности $W(U_{\text{с эфф}})$ и $W(U_{\text{п эфф}})$ соответственно.

Плотности вероятности $W(U_{\text{с эфф}})$, $W(U_{\text{п эфф}})$, в соответствии с работой [14], опишем логарифмически нормальным законом и представим выраженные в децибелах относительно 1 мкВ значения $U_{\text{с эфф}}$ и $U_{\text{п эфф}}$ как случайные величины в виде следующих выражений:

$$W(U_{\text{с эфф}}) = \frac{1}{\sqrt{2\pi}\cdot\sigma_{U_{\text{с эфф}}}} \exp\left(-\frac{(U_{\text{с эфф}} - \overline{U_{\text{с эфф}}})^2}{2\sigma^2_{U_{\text{с эфф}}}}\right); \tag{3}$$

$$W(U_{\text{п эфф}}) = \frac{1}{\sqrt{2\pi}\cdot\sigma_{U_{\text{п эфф}}}} \exp\left(-\frac{(U_{\text{п эфф}} - \overline{U_{\text{п эфф}}})^2}{2\sigma^2_{U_{\text{п эфф}}}}\right), \tag{4}$$

где $\overline{U_{\text{с эфф}}}$ и $\overline{U_{\text{п эфф}}}$, $\sigma_{U_{\text{с эфф}}}$ и $\sigma_{U_{\text{п эфф}}}$ — средние значения и среднеквадратические отклонения уровней сигналов и помех соответственно.

К сожалению реализация процедур рассмотренного метода связана со снижением показателей быстродействия РТС по сравнению с режимом непосредственного использования выделенных частотных каналов. Так, в работе [15] приведены теоретические подходы к оценке данных показателей. В частности, обосновано, что с увеличением массива непригодных частот приводит почти к двухкратному возрастанию времени, затрачиваемого на передачу информационных пакетов фиксированной длительности. При этом наибольшее абсолютное возрастание данного показателя происходит в условиях равенства длины тестовой последовательности и информационного пакета.

Однако представленные ограничения получены в условиях однородных параметров радиоканала на рабочих частотах. В условиях неоднородности распределений вида (3) и (4) как моделей каналов на каждой из выделенных частот применение формализованного метода управления ресурсами РТС в режиме *FHSS* приведет к дополнительному снижению эффективности вследствие изменения статистических характеристик радиоканалов.

Данные утверждения основываются на нарушении условий центральной предельной теоремы Ляпунова [16] ввиду ограниченного массива выделенных частот для работы РТС и появления в связи с этим зависимых случайных процессов, вносящих существенный вклад в общий процесс.

В целях дальнейшего исследования рассмотренных аспектов докажем следующее предположение:

непрерывный равномерный закон выбора рабочих частот для РТС в режиме FHSS не изменяет функцию распределения, характеризу-

ющую превышение уровня сигнала над уровнем помех на всем массиве рабочих частот

$$F_{FHSS\ \text{равн}}(z) = F_f(z),$$

где

$$F_f(z) = \int_{-\infty}^{\infty} W_f(s)\mathrm{d}s,\ z = U_{\text{с эфф}}(\text{дБ}) - U_{\text{п эфф}}(\text{дБ}).$$

При технической реализации процесса зондирования выделенного диапазона в целях исключения рабочих частот, реализующего пороговый метод управления частотным ресурсом, в общем случае функция распределения превышения уровня сигнала над уровнем помех отличается от функции распределения превышения уровня сигнала над уровнем помех на рабочих частотах, причем вид данной функции зависит от размера массива отбракованных частот.

2. Статистическая модель РТС с *FHSS*

Для доказательства сформулированного предположения воспользуемся аналитическим аппаратом свертки функций вида (3) и (4) на массиве рабочих частот.

Данный подход накладывает некоторые ограничения на степень зависимости рабочих частот, что может привести к незначительному искажению конечных результатов. Поэтому для получения общего вида функций распределения частот для РТС в режиме с *FHSS* воспользуемся методами имитационного моделирования, формализовав, таким образом, статистическую модель РТС в режиме с *FHSS*.

Для обеспечения требуемой точности получаемой оценки анализируемых статистических функций $W^*_{FHSS}(z)$ сделаем допущение, что массив рабочих частот РТС составляет $m = 1000$ рабочих частот, причем плотности вероятности превышения уровня сигнала над уровнем помех на каждой из них описываются нормальными законами вида (3) и (4) с различными параметрами распределений, а такие параметры, как математические ожидания z и дисперсии σ_z указанных распределений, представляют собой случайные массивы, описываемые равномерным законом распределения в заданных границах.

Для формализации метода управления рабочие субканалы представлены вариационным рядом, в котором субканалам с большим индексом соответствуют значения с большим превышением уровня сигнала над уровнем помех:

$$z_1 \leqslant z_2 \leqslant \ldots \leqslant z_{1000}. \qquad (5)$$

Имитационная модель РТС в режиме с *FHSS* на основе метода отбраковки рабочих частот была реализована в среде MATLAB.

На основе разработанной имитационной модели и в соответствии с известными законами распределения уровней сигналов и помех на рабочих частотах *W*f* (*z*) проведено моделирование сигнальной и помеховой обстановки на массиве *W*f* (*z*). На следующих этапах имитировалось функционирование РТС в заданных условиях, на основе которого и проводился набор статистики превышений уровня сигнала над уровнем помех. Полученные массивы данных обрабатывали и с использованием нормировок рассчитывали статистические функции распределения в каждом из радиоканалов.

Обобщенный программный код ядра имитационной модели обработки массивов данных представлен ниже в виде технологических процедур.

```
%Исходные данные для моделирования
l1 = 500; %граница порога неиспользуемых частот
l2 = 1000; %общее число частот
n = 10000; %число измерений отношения сигнал/
шум на одной частоте
%задание значений параметров статистических
характеристик законов
%распределения помех на каждой частоте
b1 = random('uniform',0.5,3.5,l2,1);
m1 = random('uniform',22,28,l2,1);
for i = 1:l2
    b(i,1:n) = b1(i);
    m(i,1:n) = m1(i);
end
%создание массива значений отношения сигнал/
шум на каждой частоте,
%распределенных по нормальному закону
z = normrnd(m,b,[l2,n]);
%представление значений отношений сигнал/шум
вариационным рядом,
%в котором частотам с большим индексом
соответствует большее превышение
%уровня сигнала над уровнем помех
z = sort(z);
%отбраковка частот со значением сигнал/шум,
не удовлетворяющим требуемому
%значению
z = z(l1 + 1:l2,:);
%создание массива значений сигнал/шум, исполь-
зуемых при моделировании линии
%ППРЧ
z = z(:);
%построение статистической функции распре-
деления в радиоканале.
[w1,w2] = hist(z,100);
w4 = w1./((l2-l1)*n);
grid on
hold on
plot(w2,w4);
```

В представленной модели происходит формирование массива из 1000 случайных величин, распределенных по закону, определяемому выражениями (3) и (4) с различными значениями параметров математического ожидания m и дисперсии b. После выполнения данной процедуры осуществляется построение вариационного ряда в соответствии с формулой (5). На основе границ отбраковки частот формируется массив случайных чисел, который и определяет режим функционирования РТС. Расчет статистических характеристик РТС осуществлялся по данным сформированного массива.

Реализация разработанной имитационной модели выбора рабочих частот для РТС в режиме с *FHSS* осуществлялась с использованием процедур *Simulink* с построением графиков состояний в фиксированные моменты времени, определяемые параметрами *FHSS* с учетом структурно-функциональных особенностей РТС.

3. Результаты эксперимента

Для подтверждения сформулированного предположения был проведен эксперимент, результаты которого приведены на рис. 1—4. При этом для удобства их сравнения использованы одинаковые масштабирующие коэффициенты для их отображения в единой системе координат.

Так, на рис. 1 представлены результаты моделирования при отбраковке $n = 10, 50, 100$ частот соответственно при общем их числе $m = 1000$. Анализ полученных результатов показывает, что при заданных параметрах управления частотно-временным ресурсом РТС в режиме с *FHSS*, определяемом отбраковкой не более 10 % частот, общий вид функции распределения превышения уровня сигнала над уровнем помех может быть аппроксимирован законами распределения вида (3) и (4), с параметрами, определяемыми центральной предельной теоремой Ляпунова.

На рис. 2 приведены статистические характеристики РТС при отбраковке соответственно $n = 250, 300, 400$ рабочих частот.

Для рассматриваемых условий при отбраковке не более 40 % частот изменение функции распределения превышения уровня сигнала над уровнем помех может быть аппроксимировано на основе моделей (1) и (2).

При отбраковке $n = 700, 800, 900$, характеризующих исключение от 40 до 90 % от общего массива рабочих частот, оставшиеся каналы

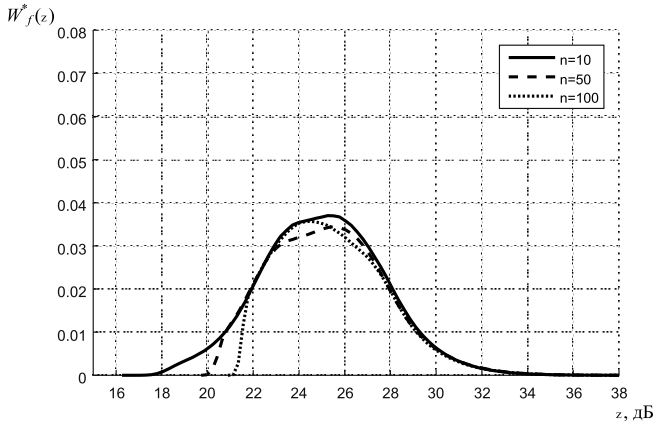


Рис. 1. Функции плотности вероятности превышения уровня сигнала над уровнем помех при исключении $n = 10, 50, 100$ рабочих частот

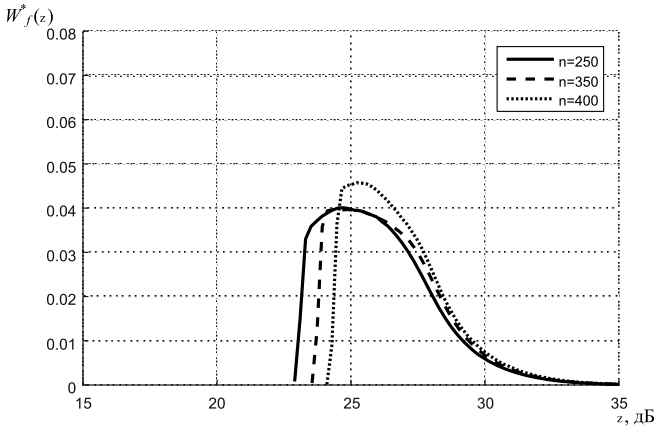


Рис. 2. Функции плотности вероятности превышения уровня сигнала над уровнем помех при исключении $n = 250, 350, 400$ рабочих частот

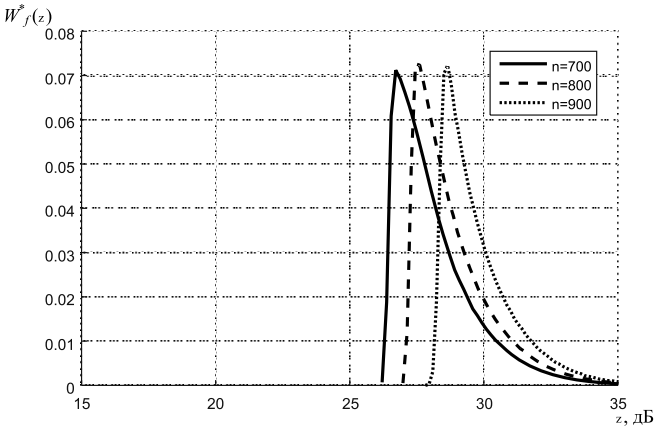


Рис. 3. Функции плотности вероятности превышения уровня сигнала над уровнем помех при исключении $n = 700, 800, 900$ рабочих частот

аппроксимируются функцией распределения экспоненциального закона (рис. 3).

Дальнейшая отбраковка массива частот (более 90 %) описывается нормальным законом

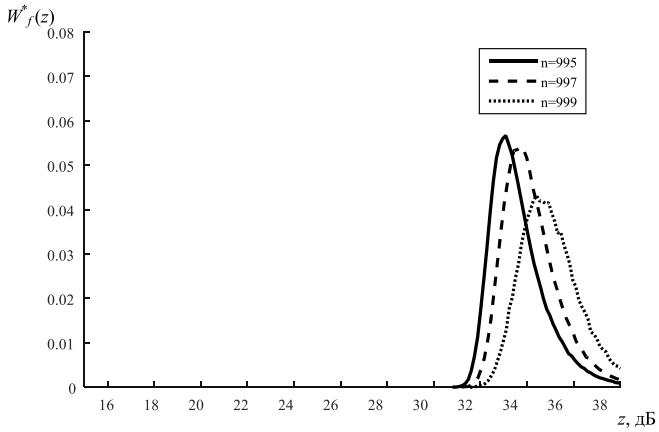


Рис. 4. Функции плотности вероятности превышения уровня сигнала над уровнем помех при исключении n = 995, 997, 999 рабочих частот

распределения статистических характеристик РТС (рис. 4), параметры которого могут быть определены на основе центральной предельной теоремы Ляпунова.

Таким образом, результаты имитационного моделирования полностью подтверждают сформулированное предположение.

Заключение

Полученные результаты имитационного моделирования подтверждают и обосновывают характер изменения статистических характеристик РТС в режиме с *FHSS* с пороговым методом управлением частотным ресурсом, что определяет их использование при расчете показателей достоверности и своевременности передачи информации в организуемых на их основе каналах радиосвязи. Следует отметить, что функции распределения экспоненциального вида характеризуют наибольшее снижение заданных показателей эффективности, соответствующее поражению более 50 % частотного ресурса.

В заключение следует отметить, что управление частотно-временным ресурсом РТС в режиме с *FHSS* на основе зондирования и отбраковки рабочих частот в моделях радиоканалов с неоднородными параметрами наиболее эффективно в случае, когда более 40 % от общего массива частот поражено в результате ПДВ. Данный факт подтверждает выводы теоретического исследования в работе [15].

Заметим, что практическая реализация РТС с управлением их ресурсами предполагает дальнейшую оценку эффективности ее алгоритмов функционирования. При этом известные методики оценки, приведенные в частности в работах [3, 7, 17], не учитывают данный аспект, что существенно ограничивает их практическое применение. Таким образом, можно полагать, что результаты имитационного моделирования открывают новое направление в решении целого класса задач, связанного с оценкой эффективности РТС со сложными моделями управления их ресурсами.

Список литературы

1. **Jakubik T., Jenicek J.** Asymmetric low-power FHSS algorithm // IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM). Donostia. San Sebastian. Spain. 2017. P. 1—6.

2. **Motlagh N. H.** Frequency Hopping Spread Spectrum: An Effective Way to Improve Wireless Communication Performance. Advanced Trends in Wireless Communications. 2011. 221 p.

3. **Борисов В. И., Зинчук В. М., Лимарев А. Е.** Помехозащищенность систем радиосвязи расширением спектра сигналов методом псевдослучайной перестройки рабочей частоты. М.: РадиоСофт, 2008. 512 с.

4. **Каунов А. Е., Поддубный В. Н.** Воздействие различных видов заградительных помех на линию радиосвязи с ППРЧ // Радиотехника. 2006. № 6. С. 58—63.

5. **Frédéric Maussang, Benjamin Ollivier, René Garello.** On the use of a FHSS modulated signal in multi-users underwater detection context // OCEANS 2017. Aberdeen. 2017. P. 1—4.

6. **Andjela Draganić, Irena Orović, Srdjan Stanković.** Spread-spectrum-modulated signal denoising based on median ambiguity function // ELMAR 2017 International Symposium. 2017. P. 63—66.

7. **Борисов В. И., Лимарев А. Е., Лепендин А. В., Маркин В. Г., Шестопалов В. И., Чаркин Д. Ю.** Вероятность ошибки на бит при множественном доступе в сетях с ППРЧ // Теория и техника радиосвязи. 2015. № 4. С. 36—46.

8. **Дворников С. В., Пшеничников А. В.** Формирование спектрально-эффективных сигнальных конструкций в радиоканалах передачи данных контрольно-измерительных комплексов // Известия высших учебных заведений. Приборостроение. 2017. Т. 60, № 3. С. 221—228.

9. **Povalac A., Sebesta J.** Phase of arrival ranging method for UHF RFID tags using instantaneous frequency measurement // ICECom, Conference Proceedings, 2010.

10. **Пшеничников А. В.** Интегральная модель радиолинии в конфликтной ситуации // Информация и космос. 2016. № 4. С. 39—45.

11. **Marko Hoyhtya.** Adaptive power and frequency allocation strategies in cognitive radio systems // Espoo 2014. VTTScience 61. P. 115—119

12. **Дворников С. В., Русин А. А., Пшеничников А. В.** Обобщенная функциональная модель радиолинии с управлением ее частотным ресурсом // Вопросы радиоэлектроники. Серия: Техника телевидения. 2016. № 3 (26). С. 49—56.

13. **Отчет** МСЭ-R F.2061 Системы фиксированной ВЧ радиосвязи, 2006.

14. **Комарович В. Ф., Сосунов В. Н.** Случайные радиопомехи и надежность КВ-связи. М.: Связь, 1977. 136 с.

15. **Дворников С. В., Пшеничников А. В., Гордейчук А. Ю.** Адаптивный выбор частот в многоканальных системах передачи видео // Вопросы радиоэлектроники. Серия: Техника телевидения. 2018. № 4. С. 68—74.

16. **Вентцель Е. С., Овчаров Л. А.** Теория вероятностей и ее инженерные приложения. 2-е изд. М.: Высшая школа, 2000. 480 с.

17. **Чаркин Д. Ю., Алехин С. Ю., Григорьев Е. В., Лимарев А. Е., Прохоров В. Е.** Оценка помехоустойчивости гибридных ППРЧ-ШПС систем радиосвязи // Теория и техника радиосвязи. 2018. № 4. С. 85—91.

S. V. Dvornikov, Dr. Tech. sciences, prof., e-mail: practicdsv@yndex.ru,
A. V. Pshenicnikov, Dr. Tech. Sciences, Assoc., e-mail: siracooz77@mail.ru,
S. S. Manaenko, Ph.D., Art. prep., e-mail: manaenkoss@mail.ru,
St. Petersburg Military Academy of Communications

Statistical Model of Interference-Free Radio Systems Based on the Threshold Frequency Resource Control Method

One of the important problems of building modern radio systems is the fulfillment of the requirements for the stability of their functioning in difficult conditions of signal and interference conditions. The greatest relevance of the task of this subject area is acquired when it is necessary to take into account the parameters of deliberate destructive influences. Known approaches are based on expanding the base of signals based on the resources of radio engineering systems. Such approaches are quite effective in conditions of homogeneous models of radio channels, which are characterized by the same parameters of the functions of the distribution of signal characteristics and interference at operating frequencies. Under the conditions of inhomogeneous models of radio channels, the selected class of solutions is rather limited due to the uncertainty of the parameters of the statistical models of radio systems. The aim of the work is to study the properties of the statistical model of radio systems that implement the threshold method of controlling the frequency resource. This method is based on the implementation of procedures for sounding working channels, determining the signal level exceeding the noise level and eliminating working frequencies with unacceptable values of the measured parameter. To achieve the stated goal, a simulation model of noise-free radio engineering systems was developed based on the threshold frequency resource control method. A hypothesis is formulated, based on the dependence of statistical laws on the parameters of the array of excluded frequencies. To prove the hypothesis, an experiment was conducted using the developed simulation model. A set of statistics was carried out, on the basis of which the characteristics of the studied radio engineering systems were obtained. Analysis of the results confirms the validity of the formulated hypothesis. The main conclusions of the work confirm and justify the nature of changes in the statistical characteristics of noise-free radio engineering systems with a threshold method of frequency resource control. At the same time, the distribution functions of the exponential type characterize the largest decrease in the specified performance indicators. In conclusion, shows the approximate boundaries of the array of excluded frequencies, defining the nature of changes in statistical parameters. (не менее 200 слов).

Keywords: *interference-free radio links, threshold control method, frequency resource, signal and interference level, statistical model*

DOI: 10.17587/it.26.16-21

References

1. **Jakubik T., Jenicek J.** Asymmetric low-power FHSS algorithm, *IEEE International Workshop of Electronics, Control, Measurement, Signals*, Donostia, San Sebastian, Spain, 2017, pp. 1—6.

2. **Motlagh N. H.** Frequency Hopping Spread Spectrum: Improving Wireless Communication Performance, Advanced Trends in Wireless Communications, 2011, 221 p.

3. **Borisov V. I., Zinchuk V. M., Limarev A. E.** Interference immunity of radio communication systems by extending the spectrum of signals by the method of pseudo-random tuning of the operating frequency, Moscow, RadioSoft, 2008, 512 p. (in Russian).

4. **Kaunov A. Ye., Poddubny V. N.** The impact of various types of barrage interference on a radio link with frequency hopping, *Radiotekhnika*, 2006, no. 6, pp. 58—63 (in Russian).

5. **Maussang F., Ollivier B., Garello R.** On the use of the FHSS modulated signal in the multi-users underwater detection context, *OCEANS 2017*, Aberdeen, 2017, pp. 1—4.

6. **Draganić A., Orović I., Stanković S.** Spread-spectrum-modulated signal denoising based on median ambiguity function, *ELMAR 2017 International Symposium,* 2017, pp. 63—66.

7. **Borisov V. I., Limarev A. E., Lependin A. V., Markin V. G., Shestopalov V. I., Charkin D. Yu.** Probability of error per bit in case of multiple access in networks with frequency hopping, *Teoriya i tekhnika radiosvyazi*, 2015, no. 4, pp. 36—46 (in Russian).

8. **Dvornikov S. V., Pshenichnikov A. V.** Formation of Spectral-Effective Signal Constructions in Radio Data Channels of Measuring Complexes, *Izvestiya Vysshikh Uchebnykh Zavedenii. Instrument making*, 2017, vol. 60, no. 3, pp. 221—228 (in Russian).

9. **Povalac A., Sebesta J.** UHF RFID tagging method using instantaneous frequency measurement, *ICECom*, *Conference Proceedings*, 2010.

10. **Pshenichnikov A. V.** Integral model of a radio link in a conflict situation, *Informatsiya i Kosmos*, 2016, no. 4, pp. 39—45 (in Russian).

11. **Hoyhtya M.** Adaptive power systems and radio systems, *Espoo 2014*, *VTTScience 61*, pp. 115—119.

12. **Dvornikov S. V., Pshenichnikov A. V., Rusin A. A.** Generalized functional model of a radio link control its frequency recource, *Voprosy Radioelektroniki. Seriya: Tekhnika Televideniya*, 2016, no. 3, pp. 49—56 (in Russian).

13. **Report** ITU-R F.2061 Fixed HF radiocommunication systems, 2006 (in Russian).

14. **Komarovich V. F., Sosunov V. N.** Random radio interference and reliability of HF communication, Moscow, Svyaz', 1977, 136 p. (in Russian).

15. **Dvornikov S. V., Pshenichnikov A. V., Gordeychuk A. Yu.** Adaptive Frequency Selection in Multichannel Video Transmission Systems, *Voprosy Radioelektroniki. Seriya: Tekhnika Televideniya*, 2018, no. 4, pp. 68—74 (in Russian).

16. **Wentzel, E. S., Ovcharov, L. A.** Theory of Probability and its Engineering Applications, 2nd ed., Moscow, Vysshaya shkola, 2000, 480 p. (in Russian).

17. **Charkin D. Yu., Alekhin S. Yu., Grigoriev Ye. V., Limarev A. Ye., Prokhorov V. Ye.** Evaluation of the noise immunity of hybrid frequency hopping circuits, radio communication systems, *Teoriya i Tekhnika Radiosvyazi*, 2018, no. 4, pp. 85—91 (in Russian).

УДК 004.722.23 DOI: 10.17587/it.26.22-29

А. Ю. Романов, канд. техн. наук, доц., e-mail: a.romanov@hse.ru,
М. В. Сидоренко, студент, e-mail: mvsidorenko@edu.hse.ru,
Национальный исследовательский университет "Высшая школа экономики", г. Москва,
Э. А. Монахова, канд. техн. наук, доц., ст. науч. сотр., e-mail: emilia@rav.sscc.ru,
Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск

Маршрутизация в сетях-на-кристалле с топологией трехмерный циркулянт

Представлена реализация динамического алгоритма маршрутизации, предназначенного для использования в сетях-на-кристалле с топологией трехмерный циркулянт (размерности 3). По сравнению с классическими алгоритмами A или Дейкстры предложенный алгоритм не требует рассчитывать весь путь прохождения пакета, а проводит расчет номера порта, в который надо направить пакет, чтобы он гарантированно достиг узла назначения. Алгоритм может быть реализован в виде цифрового автомата для выбора маршрута, что позволяет значительно упростить структуру маршрутизаторов в сетях-на-кристалле.*

Ключевые слова: *сеть-на-кристалле, алгоритм Дейкстры, циркулянт размерности 3, трехмерный циркулянт, алгоритм маршрутизации в трехмерных циркулянтах*

Введение

В настоящее время одним из важнейших направлений исследований в области информатики и вычислительных систем является построение многоядерных процессоров. Переход к многоядерным процессорам позволяет преодолеть снижение производительности при проектировании все более сложных одноядерных систем [1]. В условиях растущего интереса к технологиям построения систем-на-кристалле (System-on-Chip, SoC) и мультипроцессорных систем-на-кристалле (Multi-Processor System-on-Chip, MPSoC) [2—4] приобретают широкое распространение сети-на-кристалле (Network-on-Chip, NoC, СтнК) [5—7].

Одной из актуальных проблем в исследовании СтнК является поиск оптимальных топологий, поскольку классические регулярные топологии (mesh [8], torus [9], hypercube [10], spidergon [11]) не удовлетворяют современным требованиям к сетям на кристалле, особенно с увеличением числа узлов [12]. К таким требованиям можно отнести высокую масштабируемость и независимое от топологии число узлов сети (число узлов не обязательно должно являться степенью какого-либо числа, либо простым числом, как, например, для топологий mesh и torus). Попытка сохранения основных характеристик таких топологий приводит к большим затратам ресурсов. Сравнительно недавно в ряде работ [13, 14] было предложено для реализации сетей-на-кристалле использовать топологии циркулянтов. Достичь приемлемых значений приведенных выше параметров для циркулянтов с тремя образующими позволяет их высокая степень связности и относительно небольшое среднее расстояние между узлами [15, 16].

Циркулянт — это неориентированный граф, состоящий из множества вершин и множества образующих. Образующие циркулянта — это фиксированные числа, составляющие конфигурацию графа. По ребрам, соответствующим образующим, осуществляется переход из одной вершины в другую, и таким образом формируется маршрут из начальной вершины в конечную. Циркулянтные топологии имеют лучшие характеристики по сравнению со стандартными топологиями, например, гиперкубами: они обладают показателями лучшей структурной живучести, надежности и связности, а также требуют меньшего числа межпроцессорных обменов при решении вычислительных задач и задач системного управления [14]. Данные ха-

рактеристики обусловлены структурой самого циркулянтного графа и его симметричностью. Это позволяет использовать циркулянты в сетях с большим числом узлов, насчитывающим десятки и сотни вычислительных узлов, что уже сейчас с появлением систем с 48, 80 и больше ядрами [17] является насущной необходимостью. Ряд известных семейств циркулянтов хорошо описан в научной литературе, например, рекурсивные циркулянты [18], а также их подвиды — мультипликативные циркулянты [19], кольцевые циркулянты и другие. Для некоторых типов циркулянтов (размерности 2) известны формулы для нахождения оптимальных циркулянтов, т. е. циркулянтов с минимальным диаметром (диаметр графа — это наибольшее расстояние между всеми парами вершин графа), для других необходима разработка программных средств для их поиска. Применительно к топологии сети граф должен быть оптимальным, чтобы маршруты из одной вершины в другую имели меньшую длину. Само понятие "оптимального графа" определяется с точки зрения реализации структуры графа.

Для применения циркулянтов в качестве топологий для сетей-на-кристалле важно разработать алгоритмы маршрутизации в них, учитывающие особенности данных семейств графов и организации сетей на кристалле. В связи с тем что топология циркулянтов обладает приведенными выше характеристиками, значительно превосходящими характеристики графов с другими топологиями, можно предположить, что их использование будет более эффективным, чем остальных, однако для более точной оценки преимущества таких графов необходимо разработать и провести анализ алгоритмов маршрутизации в таких графах применительно к сетям-на-кристалле.

1. Циркулянты с тремя образующими

Циркулянтной сетью называется неориентированный граф $C(N; s_1, s_2, ..., s_k)$, где $1 \leqslant s_1 < s_2 < ... < s_k < N$ — целые числа, имеет N вершин, перенумерованных числами 0, 1, ..., $N - 1$, и каждая вершина с номером i связана ребрами с $2k$ вершинами с номерами $i + s_j (\mathrm{mod} N)$ и $i - s_j (\mathrm{mod} N)$ для всех j, $1 \leqslant j \leqslant k$. Числа s_1, s_2, ..., s_k называют образующими циркулянтного графа, число его образующих — размерностью графа. Размерность графа равна полустепени его вершин. Здесь и далее рассматриваются циркулянтные графы четной степени. На рис. 1 (см. третью сторону обложки) изображены примеры циркулянтов размерности три.

Основной характеристикой циркулянтного графа является его диаметр. Диаметром графа C называется число $d(C) = \max\limits_{i,j \in V} \mathrm{len}(i, j)$, где $\mathrm{len}(i, j)$ — длина наименьшего пути из вершины i в вершину j графа C; $V = \{1, ..., N\}$ — множество вершин графа [20]. В связи с необходимостью минимизации данной характеристики существует фундаментальная проблема теории графов — синтез оптимальных графов, т. е. таких графов, диаметр которых минимален.

В настоящей работе рассматриваются неориентированные циркулянты типа $C(N; s_1, s_2, s_3)$. Они имеют три типа ребер (s_1, s_2 и s_3). Из каждой вершины выходит по паре ребер каждого типа, симметричных относительно линии, разделяющей граф пополам из вершины, откуда исходит ребро (линия является биссектрисой угла, образуемого парой самых длинных образующих). Примерами циркулянтов такого типа могут служить графы, приведенные на рис. 1.

2. Разработка алгоритма маршрутизации для сетей-на-кристалле с топологией трехмерный циркулянт

Как было отмечено ранее, необходима разработка алгоритма маршрутизации в трехмерных циркулянтах. Типичный подход — использовать алгоритмы, подобные алгоритму Дейкстры [21]. В таком случае на каждом узле на каждом шаге пакета надо будет рассчитывать все возможные пути для достижения узла назначения, хранить таблицы маршрутизации в узлах или путь в пакете, предварительно его рассчитав. Все эти способы достаточно ресурсозатратны, поэтому необходим алгоритм, который бы позволил на каждом узле вычислить следующий шаг для движения пакета на пути в узел назначения.

Рассмотрим циркулянт типа $C(N; s_1, s_2, s_3)$. Пусть s_1, s_2, s_3 — образующие, упорядоченные по возрастанию, причем если хотя бы для одного s_i ($i = 1, ..., 3$) не выполняется неравенство $s_i < [N/2]$, где квадратные скобки обозначают целую часть от деления, тогда, если $N - s_i \geqslant 0$, то значение образующей становится равной $s_i = N - s_i$.

Из любой вершины графа можно перейти в шесть других вершин по ребрам, соответствующим значениям образующих s_1, s_2, s_3, как в направлении часовой стрелки, так и в направлении против часовой стрелки (рис. 2, см. тре-

тью сторону обложки), пройдя по s_1, s_2, s_3 и $-s_1$, $-s_2$, $-s_3$ соответственно.

Таким образом, при переходе в следующую вершину по одному из ребер, соответствующих образующим, из начальной вершины для поиска дальнейшего пути становятся доступными только пять образующих, так как по одному ребру уже был осуществлен переход в текущую вершину.

Пусть далее N_1 — текущая вершина, N_2 — конечная. Для того, чтобы определить, по какому из ребер двигаться дальше, в алгоритме при каждом изменении вершины рассчитываются четыре расстояния:

$$L_1 = \min(|N_2 - N_1|, N - |N_2 - N_1|);$$
$$L_2 = \max(|N_2 - N_1|, N - |N_2 - N_1|) = N - L_1;$$
$$L_3 = N + L_1;$$
$$L_4 = N + L_2,$$

где L_1 — минимальное расстояние от N_1 до N_2; L_2 — максимальное расстояние от N_1 до N_2 (при движении в другую сторону); L_3 — расстояние L_1 после полного обхода (цикла) вокруг графа в сторону наименьшего расстояния от N_1 до N_2; L_4 — расстояние L_2 после полного обхода вокруг графа в сторону наибольшего расстояния от N_1 до N_2.

Если одно из четырех расстояний от каждой из пяти вершин (в начальной вершине N_1 — из шести), в которые можно перейти по ребрам единичной длины, соответствующим образующим единичной длины, делится на одну из образующих нацело, то устанавливается "приоритет" частному — "high"; если расстояние делится на сумму образующих (некоторые члены суммы могут не входить в сумму, например $s_2 + s_1$, или быть со знаком минус, например $s_1 - s_2 + s_3$), то устанавливается приоритет частному "low". Само частное умножается на 2 или на 3 для случаев суммы двух или трех элементов, так как потребуется в 2 или 3 раза больше шагов алгоритма соответственно.

Среди всех частных выбирается минимальное и сохраняется в паре с соответствующей ему образующей. Среди всех рассчитанных пар выбирается пара с наивысшим приоритетом, которой для достижения конечной вершины N_2 требуется минимальное число шагов. В данной паре результатом является образующая s_i, взятая со знаком плюс или минус в зависимости от того, по какому из ребер был осуществлен переход в текущую вершину.

Для того чтобы избежать зацикливания, в алгоритме проверяется вхождение текущей вершины в список уже "посещенных" вершин. Если вершина входит в данный список, следующий шаг происходит уже по одной из тех образующих, которые не равны текущей.

На основе приведенных выше утверждений разработан следующий алгоритм:

algorithm Find_Route_Sequent is

Input: N_1 — start node, N_2 — end node, N — count of nodes, s_1 — first generatrix, s_2 — second generatrix, s_3 — third generatrix, *previosStep* — last algorithm step (0 by default), *theFirst* — start vertex in the graph.

Output: N_1 — next start node.

```
1:  sList ← [s1, s2, s3, −s1, −s2, −s3]
2:  removeFlag ← false
3:  if −previousStep in sList then
4:  removePosition ← sList.IndexOf(−previousStep)
5:  delete sList[−previousStep]
6:  removeFlag ← true
7:  x ← [N1 + s1, N1 + s2, N1 + s3, N1 − s1, N1 − s2, N1 − s3]
8:  if removeFlag then
9:  delete x[removePosition]
10: distances ← [ ]
11: for i ← 0; i < x.Count; i ← i + 1 do
12: if x[i] ≤ 0 then
13: x[i] ← x[i] + N
14: if x[i] > N then
15: x[i] ← x[i] mod N
16: distances ← [
17: min(|N2 − N1|, N − |N2 − N1|),
18: max(|N2 − N1|, N − |N2 − N1|),
19: N + min(|N2 − N1|, N − |N2 − N1|),
20: N + max(|N2 − N1|, N − |N2 − N1|)
21: ]
22: if theFirst in x then
23: removeTfPos ← x.IndexOf(theFirst)
24: delete sList[removeTfPos]
25: delete x[removeTfPos]
26:
27: minDistances ← [ ], minLengths ← [ ]
28: for i ← 0; i < distances.Count; i ← i + 1 do
29: minDistances.Clear()
30: d ← distances[i]
31: for idx ← 0; idx < d.Count; idx ← idx + 1 do
32: if d[idx] mod s3 = 0 then
33:         minDistances.Add(d[idx] div s3, sList[i], "high")
34: if d[idx] mod s2 = 0 then
35:         minDistances.Add(d[idx] div s2, sList[i], "high")
36: if d[idx] mod s1 = 0 then
37:         minDistances.Add(d[idx] div s1, sList[i], "high")
38: if d[idx] mod (s1 + s2) = 0 then
39:         minDistances.Add(d[idx] div(s1 + s2)*2, sList[i], "low")
40: if d[idx] mod (s1 + s3) = 0 then
```

```
        minDistances.Add(d[idx] div(s1 + s3)*2, sList[i], "low")
if d[idx] mod (s2 + s3) = 0 then
        minDistances.Add(d[idx] div(s2 + s3)*2, sList[i], "low")
if d[idx] mod (s2 − s1) = 0 . then
        minDistances.Add(d[idx] div(s2 − s1)*2, sList[i], "low")
if d[idx] mod (s3 − s2) = 0 then
        minDistances.Add(d[idx] div(s3 − s2)*2, sList[i], "low")
if d[idx] mod (s3 − s1) = 0 then
        minDistances.Add(d[idx] div(s3 − s1)*2, sList[i], "low")
if d[idx] mod (s1 + s2 + s3) = 0 then
        minDistances.Add(d[idx] div(s1 + s2 + s3)*3, sList[i], "low")
if d[idx] mod (s1 − s2 + s3) = 0 then
        minDistances.Add(d[idx] div(s1 − s2 + s3)*3, sList[i], "low")
if d[idx] mod (−s1 + s2 + s3) = 0 then
        minDistances.Add(d[idx] div(−s1 + s2 + s3)*3, sList[i], "low")
if s1 + s2 − s3 > 0 and d[idx] mod (s1 + s2 − s3) = 0 then
        minDistances.Add(d[idx] div(s1 + s2 − s3)*3, sList[i], "low")
if −s1 − s2 + s3 > 0 and d[idx] mod (−s1 − s2 + s3) = 0 then
        minDistances.Add(d[idx] div(−s1 − s2 + s3)*3, sList[i], "low")
minLenghts.Add(GetMinLength(minDistances))
resultGen ← min(minLengths, argument = 0)[1]
if resultGen = 0 then
    if prevStep > 0 then
            if s3 in sList then
            resultGen ← s3
    else if s2 in sList then
            resultGen ← s2
    else if s1 in sList then
            resultGen ← s1
else
    if −s3 in sList then
            resultGen ← −s3
    else if −s2 in sList then
            resultGen ← −s2
    else if −s1 in sList then
            resultGen ← −s1
return resultGen
```

function GetMinLength is

Input: *minLengths* — list of triples: (division_result, generatrix, priority).

Output: (a,b) — tuple with minimal steps to N_2 (a) and with corresponding generatrix (b).

```
minTHigh ← 10e5
minTLow ← 10e5
secArgHigh ← 0
secArgLow ← 0
for i ← 0; i < minLengths; i ← i + 1 do
    if minLengths[i].priority = "high" and
     minLengths[i].divisionResult < minTHigh then
            minTHigh ← minLengths[i].divisionResult
            secArgHigh ← minLengths[i].generator
    else
            minTLow ← minLengths[i].divisionResult
            secArgLow ← minLengths[i].generator
if minTHigh = 10e5 or minTLow < minTHigh then
    return (minTLow, secArgLow)
return (minTHigh, secArgHigh)
```

3. Тестирование разработанного алгоритма маршрутизации

Разработанному алгоритму присвоено название "Sequent". Для тестирования данного алгоритма был сгенерирован набор оптимальных циркулянтов (графов с минимальным диаметром) размерности 3 для различного числа вершин (9, 15, 16, 21 23, 25 36, 49, 64, 81 и 100) с помощью алгоритма Дейкстры [22]. Зависимость диаметра от числа вершин в циркулянтах из выборки [23] приведена на рис. 3.

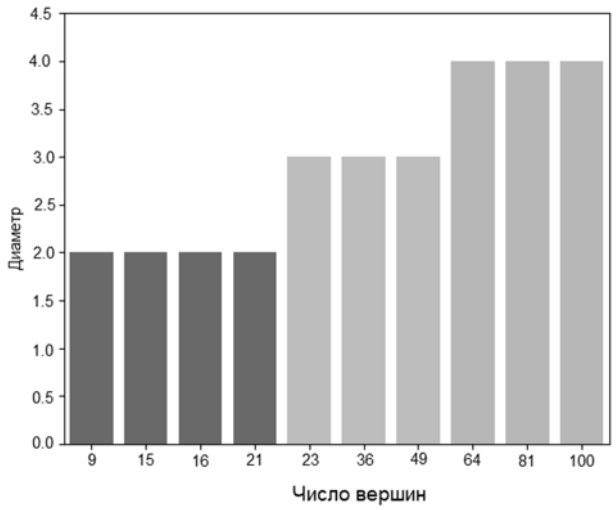


Рис. 3. Зависимость диаметра от числа вершин графов из выборки оптимальных циркулянтов

4. Оценка эффективности алгоритма "Sequent"

С использованием программных средств для синтеза циркулянтных топологий из работы [24] было сгенерировано 416 описаний оптимальных циркулянтов [23], которые были использованы для тестирования алгоритма "Sequent". На рис. 4 представлен график разностей максимальных расстояний для разработанного алгоритма с диаметром, рассчитанным по алгоритму Дейкстры [22]. Для всех отсутствующих на графике циркулянтов разность равна нулю.

Из графика следует, что в 20 из 416 тестов максимальное расстояние между узлами для алгоритма "Sequent" отличается от диаметра, а на 396 тестах — совпадает. Более того, среди

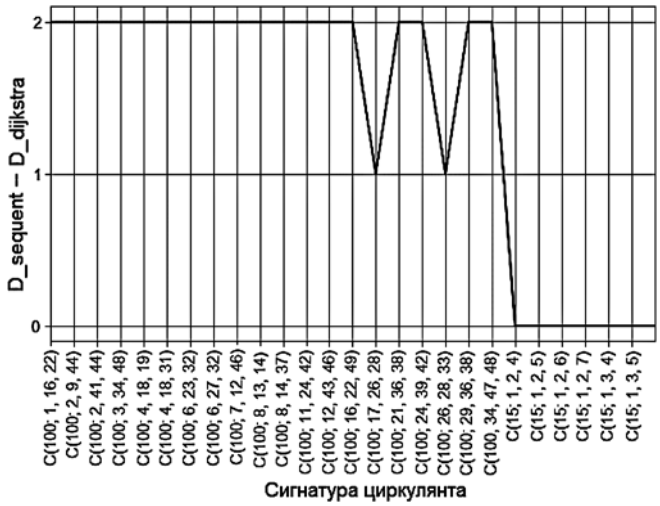


Рис. 4. График разностей максимальных расстояний для алгоритмов "Sequent" и Дейкстры (тест 1)

20 случаев разница между диаметром и рассчитанным расстоянием не превышает единицы в двух случаях и равна двум — в 18 случаях. Таким образом, эффективность алгоритма можно вычислить по формуле:

$$Alg\_efficiency = \frac{\sum_{i=1}^{416} D_{i\,Dijkstra}}{\sum_{i=1}^{416} D_{i\,Sequent}},$$

где $D_{i_{Dijkstra}}$ — диаметр i-го циркулянта, рассчитанного по алгоритму Дейкстры; $D_{i_{Sequent}}$ — максимальное расстояние между узлами для i-го циркулянта, рассчитанное по алгоритму "Sequent".

Эффективность разработанного алгоритма на основе приведенного выше набора оптимальных циркулянтов равна

$$Alg\_efficiency = \frac{1351}{1389} \approx 0,973.$$

Далее алгоритм "Sequent" был запущен для 296 циркулянтов, приведенных в работе [23]. Они распределены так, что для каждого числа вершин берется по одному оптимальному циркулянту, при этом у 96 % циркулянтов $s_1 = 1$, т. е. они являются кольцевыми. График разностей максимальных расстояний с диаметрами для данного теста приведен на рис. 5, *а* (см. третью сторону обложки).

Из графика следует, что для 209 из 296 (70 %) циркулянтов алгоритм "Sequent" дает максимальное расстояние, отличное от диаметра. Изменение разности максимального расстояния и диаметра находится в следующих пределах: 10 — для порядков графов от 5 до 154; от 3 до 20 — для порядков от 155 до 250; от 4 до 40 — для порядков от 251 до 300. Полученные в результате тестирования значения могут отличаться при повторении эксперимента из-за особенностей алгоритма. Это может происходить из-за того, что при попадании текущей вершины в список посещенных вершин, для устранения зацикливания предусмотрен переход по одному из трех ребер, отличных от того, по которому уже осуществлялся переход.

Затем алгоритм "Sequent" был реализован на выборке из 28 299 циркулянтов из набора, представленного в [23]. Результат приведен на рис. 5, *б* (см. третью сторону обложки). Исходя из полученных результатов можно сделать вывод о том, что в циркулянтах, имеющих число вершин от 300 и выше, расхождения между алгоритмом Дейкстры и алгоритмом "Sequent" становятся значительными: в 99 % случаях диаметр графа не совпадает с максимальным расстоянием, рассчитанным по разработанному алгоритму "Sequent", а среднее значение расхождения составляет 154.

Таким образом, применение данного алгоритма для сетей на кристалле с числом узлов больше 300 неэффективно. Тем не менее, порог в 300 узлов вполне достаточен для современных сетей на кристалле, где число узлов обычно не превышает 100 [17].

5. Анализ конфигураций циркулянтов, при которых алгоритм "Sequent" работает неэффективно

Анализируя устройство сетей-на-кристалле с циркулянтной топологией, можно обнаружить такие сигнатуры циркулянтов, при которых маршрут из одной вершины в другую будет слишком длинным. К таким сигнатурам относят следующий тип циркулянтов:

$$C(N;\ s_1,\ s_2,\ s_3),\ \text{при четном } N \text{ и НОД } (s_1,\ s_2,\ s_3) \neq 1,$$

где НОД (s_1, s_2, s_3) — наибольший общий делитель чисел s_1, s_2, s_3 [25].

Также возникают случаи, когда путь существует, но является большим (в десятки или сотни раз выше оптимального). К данным сигнатурам относятся следующие типы циркулянтов:

1) $C(N; s_1, s_2, s_3)$ при $s_2 = 2s_1$;

2) $C(N; s_1, s_2, s_3)$ при $s_2 - s_1 = s_3 - s_2$ и $N \bmod s_2 = 0$.

Сети-на-кристалле с таким набором вершин и образующих будут заметно менее эффектив-

ны, чем сети с циркулянтами, приведенными в работе [23].

Для наборов графов тестов 2 и 3 был проведен анализ на наличие сигнатур, для которых алгоритм "Sequent" имеет циклы и недостижимые пути. Так, для теста 2 среди общего числа циркулянтов обнаружено 2 % сигнатур, при которых алгоритм возвращает длинные маршруты, а недостижимые пути отсутствуют. В тесте 3 с числом вершин от 300 были обнаружены 285 графов с недостижимыми путями (1 % от размера выборки).

Таким образом, для реализации сетей-на-кристалле необходимо учитывать условия появления недостижимых маршрутов. Более того, для повышения эффективности работы алгоритма следует исключать длинные маршруты, применяя соответствующие условия.

6. Оценка времени выполнения алгоритма "Sequent"

Рассмотрим зависимости времени выполнения и максимального расстояния между узлами, рассчитанного по алгоритму "Sequent", от числа вершин для различных циркулянтов. Все вычисления времени выполнения алгоритма проводили на 64-разрядном компьютере с тактовой частотой 2,5ГГц с процессором Intel Core i7 и с 8 Гб оперативной памяти (рис. 6, см. третью сторону обложки).

На основе графиков на рис. 6 можно сделать вывод о том, что время выполнения алгоритма зависит от числа вершин нелинейно. Эта зависимость близка к квадратичной.

Несколько другая ситуация с максимальным расстоянием между узлами (рис. 7, см. третью сторону обложки).

Из рис. 7 следует, что зависимость максимального расстояния графа от числа вершин при увеличении числа вершин линейная. Но это не всегда так, поскольку для различных образующих при малых N зависимость отличается. Сначала она выглядит как ступенчатая функция (рис. 8, *а*, см. четвертую сторону обложки), а затем вырождается в линейную при больших N (рис. 8, *б*, см. четвертую сторону обложки).

7. Оценка сложности алгоритма "Sequent"

На каждой итерации алгоритм совершает расчет четырех расстояний для пяти вершин, в которые можно перейти по одному из пяти ребер, соответствующих образующим (для первой итерации — шесть вершин). Как было показано выше, зависимость $D(N)$ — линейная при больших $N > 100$, где D — диаметр графа. Это означает, что диаметр D можно заменить на выражение kN, где k — коэффициент, причем $0 < k < 1$. Таким образом, для одной итерации сложность алгоритма [26] составляет $O(20kN)$, что равносильно $O(N)$. Для N таких итераций сложность алгоритма равна $O(N^2)$, что не превосходит сложность вычислений по алгоритму Дейкстры [22].

В разработанном алгоритме не требуется хранить большие таблицы с данными, следовательно, использование памяти в нем минимальное.

Заключение

Разработанный алгоритм для общего случая оптимального циркулянта размерности 3 был протестирован на специально сгенерированном наборе данных из 416 оптимальных циркулянтов, и была рассчитана его эффективность. Для данного набора она равна 0,973, что является достаточным показателем эффективности алгоритма для задачи реализации алгоритма на HDL на уровне маршрутизатора сети-на-кристалле, поскольку данный алгоритм имеет линейную сложность и может быть легко описан в виде цифрового автомата.

В 95 % рассмотренных случаев для набора из графов с числом вершин, меньшим 300, алгоритм показал результат, аналогичный результату, полученному с помощью алгоритма Дейкстры. Вычислительная временная сложность разработанного алгоритма совпадает со сложностью алгоритма Дейкстры, который считается эталонным при нахождении маршрутов в сетях-на-кристалле. При числе вершин больше 300 алгоритм становится неэффективным, так как максимальное расстояние между узлами, рассчитанное по алгоритму "Sequent", может в десятки раз превышать диаметр, рассчитанный по алгоритму Дейкстры.

По сравнению с классическими алгоритмами предложенный алгоритм не требует рассчитывать весь путь прохождения пакета, а определяет номер порта, в который надо направить пакет, чтобы он гарантированно достиг узла назначения. Это позволяет значительно упростить структуру маршрутизатора СтнК.

Публикация подготовлена в ходе проведения исследования (№ 18-01-0074) в рамках Про-

граммы "Научный фонд Национального исследовательского университета "Высшая школа экономики" (НИУ ВШЭ)" в 2018—2019 гг. и в рамках государственной поддержки ведущих университетов Российской Федерации "5-100".

Исследование Монаховой Э. А. выполнено в рамках проекта ИВМиМГ СО РАН 0315-2016-0006.

Список литературы

1. **Nychis G.** et al. Next generation on-chip networks: What kind of congestion control do we need? // Proc. 9th ACM SIGCOMM Work. Hot Top. Networks. 2010. P. 1—6.

2. **Paul J.** et al. Resource-awareness on heterogeneous MPSoCs for image processing // J. Syst. Archit. 2015. Vol. 61, N. 10. P. 668—680.

3. **Wolf W., Jerraya A. A., Martin G.** Multiprocessor system-on-chip (MPSoC) technology // IEEE Trans. Comput. Des. Integr. Circuits Syst. 2008. Vol. 27, N. 10. P. 1701—1713.

4. **Phanibhushana B., Kundu S.** Network-on-chip design for heterogeneous multiprocessor system-on-chip // Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI. 2014. P. 486—491.

5. **Abdelfattah M. S., Bitar A., Betz V.** Design and Applications for Embedded Networks-on-Chip on FPGAs // IEEE Trans. Comput. 2017. Vol. 66, N. 6. P. 1008—1021.

6. **Benini L., De Micheli G.** Networks on Chip // Networks on Chips. 2006.

7. **Jantsch A.** Models of computation for networks on chip // Proceedings — International Conference on Application of Concurrency to System Design, ACSD. 2006.

8. **Deb D.** et al. Cost effective routing techniques in 2D mesh NoC using on-chip transmission lines // J. Parallel Distrib. Comput. Academic Press, 2019. Vol. 123. P. 118—129.

9. **Ansari A. Q., Ansari M. R., Khan M. A.** Modified quadrant-based routing algorithm for 3D Torus Network-on-Chip architecture // Perspect. Sci. 2016. Vol. 8. P. 718—721.

10. **Marvasti M. B., Szymanski T. H.** The performance of hypermesh NoCs in FPGAs // IEEE International Conference on Computer Design: VLSI in Computers and Processors. 2012. P. 492—493.

11. **Bishnoi R.** et al. Distributed adaptive routing for spidergon NoC // 18th International Symposium on VLSI Design and Test, VDAT 2014. 2014. P. 1—6.

12. **Dally W. J., Towles B. P.** Principles and Practices of Interconnection Networks. Elsevier, 2003. 581 p.

13. **Boesch F. T., Jhing-Fa Wang, Wang J. F.** Reliable Circulant Networks with Minimum Transmission Delay // IEEE Transactions on Circuits and Systems. 1985. Vol. 32, N. 12. P. 1286—1291.

14. **Monakhova E. A.** A Survey on Undirected Circulant Graphs // Discret. Math. Algorithms Appl. World Scientific Publishing Company, 2012. Vol. 4, N. 1. P. 1250002.

15. **Kotsis G., Awiuni B. A.** Interconnection Topologies for Parallel Processing Systems // PARS Mitteilungen. 1993. Vol. 11. P. 1—6.

16. **Lu R.** Fast methods for designing circulant network topology with high connectivity and survivability // J. Cloud Comput. Journal of Cloud Computing, 2016. Vol. 5, N. 5.

17. **Comprés Ureña I. A., Riepen M., Konow M.** RCKMPI — Lightweight MPI implementation for intel's single-chip cloud computer (SCC) // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin, Heidelberg, 2011. Vol. 6960 LNCS. P. 208—217.

18. **Park J.-H., Chwa K.-Y.** Recursive Circulant: A New Topology for Multicomputer Networks // International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 1994). 1994. P. 73—80.

19. **Stojmenović I.** Multiplicative circulant networks topological properties and communication algorithms // Discret. Appl. Math. 1997. Vol. 77, N. 3. P. 281—305.

20. **Chung F. R. K.** Diameters and Eigenvalues // J. Am. Math. Soc. 2006.

21. **Dijkstra E. W.** A note on two problems in connexion with graphs // Numer. Math. 1959. Vol. 1. P. 269—271.

22. **Deng Y.** et al. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment // Appl. Soft Comput. J. 2012. Vol. 12, N. 3. P. 1231—1237.

23. **Romanov A. Y.** Optimal Circulants Dataset [Electronic resource]. URL: https://github.com/RomeoMe5/circulantGraphs/ (accessed: 21.03.2019).

24. **Romanov A. Y., Romanova I. I., Glukhikh A. Y.** Development of a Universal Adaptive Fast Algorithm for the Synthesis of Circulant Topologies for Networks-on-Chip Implementations // 2018 IEEE 38th International Scientific Conference on Electronics and Nanotechnology, ELNANO 2018. 2018. P. 110—115.

25. **Eisenbrand F.** The euclidean algorithm // Algorithms Unplugged. 2011.

26. **Fortnow L., Homer S.** Computational Complexity // Handbook of the History of Logic. 2014.

A. Yu. Romanov, PhD, Associate Professor, e-mail: a.romanov@hse.ru,
M. V. Sidorenko, Student, e-mail: mvsidorenko@edu.hse.ru,
National Research University Higher School of Economics, Moscow, Russian Federation,
E. A. Monakhova, PhD, Associate Professor, Senior Researcher, e-mail: emilia@rav.sscc.ru,
ICMMG SB RAS, Novosibirsk, Russian Federation

Routing in Networks-on-Chip with a Three-Dimensional Circulant Topology

The paper presents the implementation of a dynamic routing algorithm intended for use in networks-on-chip with a three-dimensional circulant topology of type C(N; s1, s2, s3). Compared with the classical algorithms A or Dijkstra, the proposed algorithm does not require to calculate the entire path of the packet, but calculates the port number to which the packet should be sent so that it can reach the destination node. This makes it possible to significantly simplify the structure of the NoC router.*

The algorithm can be implemented as RTL state machine in routers for NoCs for finding the shortest routes between any two network nodes. The proposed algorithm was tested on sets of optimal circulants. For this set, it is equal to 0.973 which is a sufficient indicator of efficiency of the algorithm for the task of implementing the algorithm in HDL at the level of a network-on-chip router, since this algorithm has linear complexity and can be easily implemented. In 95 % of the cases considered, for a set of graphs with the number of vertices less than 300, the algorithm showed a result similar to that obtained using the Dijkstra algorithm. The computational time complexity of the created algorithm is similar to the complexity of the Dijkstra algorithm which is considered to be the reference when finding routes in networks-on-chip. When the number of vertices is more than 300, the algorithm becomes inefficient, since the diameters, calculated by "Sequent" algorithm, can be ten times higher than the optimal diameters calculated by the Dijkstra algorithm.

Keywords: *network-on-chip, Dijkstra algorithm, circulant of the third dimension, three-dimensional circulant, routing algorithm in three-dimensional circulants*

DOI: 10.17587/it.26.22-29

References

1. **Nychis G.** et al. Next generation on-chip networks: What kind of congestion control do we need?, *Proc. 9th ACM SIGCOMM Work. Hot Top. Networks*, 2010, pp. 1—6.

2. **Paul J.** et al. Resource-awareness on heterogeneous MPSoCs for image processing, *J. Syst. Archit.*, 2015, vol. 61, no. 10, pp. 668—680.

3. **Wolf W., Jerraya A. A., Martin G.** Multiprocessor system-on-chip (MPSoC) technology, I*EEE Trans. Comput. Des. Integr. Circuits Syst.,* 2008, vol. 27, no. 10, pp. 1701—1713.

4. **Phanibhushana B., Kundu S.** Network-on-chip design for heterogeneous multiprocessor system-on-chip, *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI,* 2014, pp. 486—491.

5. **Abdelfattah M. S., Bitar A., Betz V.** Design and Applications for Embedded Networks-on-Chip on FPGAs, I*EEE Trans. Comput.*, 2017, vol. 66, no. 6, pp. 1008—1021.

6. **Benini L., De Micheli G.** Networks on Chip, *Networks on Chips*, 2006.

7. **Jantsch A.** Models of computation for networks on chip, *Proceedings — International Conference on Application of Concurrency to System Design*, *ACSD,* 2006.

8. **Deb D.** et al. Cost effective routing techniques in 2D mesh NoC using on-chip transmission lines, *J. Parallel Distrib. Comput. Academic Press*, 2019, vol. 123, pp. 118—129.

9. **Ansari A. Q., Ansari M. R., Khan M. A.** Modified quadrant-based routing algorithm for 3D Torus Network-on-Chip architecture, *Perspect. Sci.*, 2016, vol. 8, pp. 718—721.

10. **Marvasti M. B., Szymanski T. H.** The performance of hypermesh NoCs in FPGAs, I*EEE International Conference on Computer Design: VLSI in Computers and Processors,* 2012, pp. 492—493.

11. **Bishnoi R.** et al. Distributed adaptive routing for spidergon NoC, *18th International Symposium on VLSI Design and Test, VDAT 2014*, 2014, pp. 1—6.

12. **Dally W. J., Towles B. P.** Principles and Practices of Interconnection Networks. Elsevier, 2003, 581 p.

13. **Boesch F. T., Jhing-Fa Wang, Wang J. F.** Reliable Circulant Networks with Minimum Transmission Delay, *IEEE Transactions on Circuits and Systems,* 1985, vol. 32, no. 12, pp. 1286—1291.

14. **Monakhova E. A.** A Survey on Undirected Circulant Graphs, *Discret. Math. Algorithms Appl. World Scientific Publishing Company,* 2012, vol. 4, no. 1, pp. 1250002.

15. **Kotsis G., Awiuni B. A.** Interconnection Topologies for Parallel Processing Systems, *PARS Mitteilungen*, 1993, vol. 11, pp. 1—6.

16. **Lu R.** Fast methods for designing circulant network topology with high connectivity and survivability, *J. Cloud Comput. Journal of Cloud Computing,* 2016, vol. 5, no. 5.

17. **Comprés Ureña I. A., Riepen M., Konow M.** RCKMPI — Lightweight MPI implementation for intel's single-chip cloud computer (SCC), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* Springer, Berlin, Heidelberg, 2011, vol. 6960 LNCS, pp. 208—217.

18. **Park J.-H., Chwa K.-Y.** Recursive Circulant: A New Topology for Multicomputer Networks, *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 1994),* 1994, pp. 73—80.

19. **Stojmenović I.** Multiplicative circulant networks topological properties and communication algorithms, *Discret. Appl. Math.,* 1997, vol. 77, no. 3, pp. 281—305.

20. **Chung F. R. K.** Diameters and Eigenvalues, *J. Am. Math. Soc.*, 2006.

21. **Dijkstra E. W.** A note on two problems in connexion with graphs, *Numer. Math.*, 1959, vol. 1, pp. 269—271.

22. **Deng Y.** et al. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, *Appl. Soft Comput. J.*, 2012, vol. 12, no. 3, pp. 1231—1237.

23. **Romanov A. Y.** Optimal Circulants Dataset, available at: https://github.com/RomeoMe5/circulantGraphs/ (accessed: 21.03.2019).

24. **Romanov A. Y., Romanova I. I., Glukhikh A. Y.** Development of a Universal Adaptive Fast Algorithm for the Synthesis of Circulant Topologies for Networks-on-Chip Implementations, *2018 IEEE 38th International Scientific Conference on Electronics and Nanotechnology, ELNANO 2018,* 2018, pp. 110—115.

25. **Eisenbrand F.** The euclidean algorithm, *Algorithms Unplugged,* 2011.

26. **Fortnow L., Homer S.** Computational Complexity, Handbook of the History of Logic, 2014.

УДК 004.912 DOI: 10.17587/it.26.30-38

Е. В. Полицына, канд. техн. наук, доц., e-mail: kathrin.beaver@mail.ru,
С. А. Полицын, канд. техн. наук, доц., e-mail: pul_forever@mail.ru,
А. О. Касаткина, студент, e-mail: alyona.kasatkina1997@yandex.ru,
Московский авиационный институт (национальный исследовательский университет)

Создание интегрального алгоритма и инструментов автоматического реферирования текстов на русском языке

В настоящее время количество информации, представленной в текстовом виде, с каждым годом увеличивается, тем самым задача автоматической обработки текстов, а особенно сокращения их объема, становится все более актуальной. Инструменты, предназначенные для получения реферата на русском языке, в основном используют только статистический метод. Существует необходимость исследования алгоритмов реферирования для создания новых алгоритмов и инструментов, которые будут предоставлять возможность использования нескольких методов автоматического реферирования для улучшения результатов. В статье представлены результаты исследования алгоритмов экстракции, на их основе предлагается интегральный алгоритм реферирования, библиотека и сервис автоматического реферирования текстов на русском языке для обеспечения возможности использования различных методов реферирования.

Ключевые слова: *экстракция, методы автоматического реферирования, интегральный метод реферирования, библиотека и сервис автоматического реферирования текста*

Введение

С развитием информационных технологий из года в год человечество накапливает все больше информации, в том числе представленной в текстовом виде. В связи с этим все более актуально создание автоматизированных средств ее анализа и обработки. Для удобства обработки больших массивов текстов часто необходимо получать из них наиболее важную информацию в краткой форме, обозримой для беглого просмотра или быстрого анализа.

С такой проблемой сталкиваются специалисты не только в области лингвистики, но и во многих других областях, например, редакторы и писатели при подготовке и анализе новостей, ученые при поиске информации по теме исследования, технические писатели и аналитики при подготовке документации и др. В повседневной жизни людей также постоянно возникает проблема избыточности получаемой информации, начиная от новостей из СМИ или от знакомых людей до результатов выдачи поисковых систем.

Средства автоматического построения краткого представления текста очень полезны в любых сферах, где людям необходимо справиться с огромными потоками информации и быстро принять решение о том, какая информация подходит для дальнейшей работы, а какую нужно отсеять на первом этапе. Для этого применяются методы реферирования — краткого изложения текста в письменном виде с раскрытием его основного содержания по всем затронутым вопросам [1].

О важности развития этого направления свидетельствуют и факты покупки в 2013 г. компаниями Google и Yahoo стартапов Wavii и Summly соответственно, которые занимались созданием и развитием средств реферирования. Компания "Яндекс" поддерживает работы в области автоматического реферирования веб-документов с учетом запроса [2]. Востребованность автоматических средств для сокращения объема текста подтверждают и все больше набирающие популярность мобильные приложения, которые обрабатывают новостные потоки и представляют пользователю короткие рефераты на выбранные

темы, а также создание плагинов для различных браузеров, которые позволяют получить краткое содержание веб-страниц.

Методы автоматического реферирования текстов и алгоритмы расчета весов предложений

Существует два подхода к автоматическому составлению реферата [3]: *экстракция* — извлечение из исходного текста наиболее важных информационных блоков (абзацев, предложений); *абстракция* — генерация реферата с порождением нового текста, содержательно обобщающего первичный документ. Второй подход опирается на построение модели понимания текста и его синтеза на естественном языке. Понимание смысла текста — это процесс, который происходит путем установления логических связей между предметами на основе имеющихся знаний [4], что является отдельной большой и нерешенной проблемой в компьютерной лингвистике. Таким образом, для применения подхода на основе абстракции необходимо построение семантической модели текста, наличие знаний о широком круге предметных областей в формализованном виде и разработка алгоритмов оперирования этими представлениями. Помимо сложности реализации и наличия ряда концептуальных проблем понимания текста человеком времени на получение реферата, таким образом, требуется существенно больше, чем для подхода на основе экстракции, а использование сложных структур данных и необходимость подготовки и хранения дополнительных исходных данных накладывает дополнительные требования к ресурсам компьютера.

Выделяются следующие методы экстракции [3]:

1) *статистический метод*, суть которого заключается в выделении в тексте частотных слов, вычислении весов предложений с помощью суммирования частот (весов) входящих в их состав слов и включении в реферат предложений с наибольшими весами;

2) *позиционный метод*, который опирается на предположение о том, что информативность текстового блока (предложения) находится в зависимости от его позиции (места) в тексте документа;

3) *индикаторный метод*, основанный на идентификации фраз первичного документа с помощью индексации их специальными словами — маркерами, индикаторами и коннекторами.

Помимо методов реферирования существуют также алгоритмы расчета веса предложения. Вес — количественная характеристика, отражающая значимость предложения в тексте. Для его определения выделяют следующие алгоритмы:

1. Статистический алгоритм, который основан на определении веса предложения в зависимости от весов входящих в него ключевых слов [5].

2. Алгоритм на основе симметричного реферирования, который заключается в вычислении связей данного предложения с другими [6].

Инструменты автоматического реферирования текстов

Помимо разработок компаний, занимающихся развитием поисковых систем или новостных агрегаторов, существует ряд инструментов автоматического реферирования текстов: систем с веб-интерфейсом, библиотек и т.д. Большинство из них поддерживают работу только с текстами на английском языке, наиболее интересными и работоспособными из мультиязычных систем являются SweSum, MEAD. Система MEAD представляет собой библиотеку на языке Python, графический интерфейс пользователя не предоставляется. Система SweSum предназначена для автоматического реферирования текстов на различных языках (в списке доступных языков нет русского языка). Система основана на трех методах реферирования: статистическом, позиционном и индикаторном. Даже несмотря на то, что в системе применяются три метода реферирования текста, в реферат не всегда включаются все предложения, отражающие смысл текста.

Построение рефератов по текстам на русском языке поддерживают следующие системы.

1. TextAnalyst

Система TextAnalyst используется для автоматического реферирования текстов на русском языке с помощью статистического метода, т. е. наиболее значимыми считаются те предложения, которые имеют наибольший вес. Главным недостатком программы является качество реферирования текста: выбранные предложения не полностью раскрывают смысл текста, а также не связаны друг с другом логически.

2. VisualWorld

В состав системы VisualWorld входит инструмент для автоматического реферирования текстов на русском языке "Рефератор", осно-

ванный на статистическом методе, т. е. в реферат включаются предложения с наибольшим весом. Главным недостатком системы является качество получаемого реферата: отобранные предложения не полностью раскрывают смысл текста и не всегда связаны друг с другом.

3. Text Summarization

Система Text Summarization предназначена для реферирования текстов на русском и английском языках с использованием статистического метода. Основным недостатком системы является качество получаемого реферата, так как отобранные предложения не связаны друг с другом логически.

4. TextCompactor

Система TextCompactor применяется для автоматического реферирования текстов на различных языках, в том числе и на русском языке. Система основана на статистическом и позиционном методах реферирования. Несмотря на то, что в системе применяются два подхода к реферированию текста, в реферат включаются не все предложения, отражающие основную мысль текста.

5. Tools4noobs

Система Tools4noobs применяется для автоматического реферирования текстов на различных языках, включая русский язык. Для построения реферата в системе используются статистический и позиционный методы. Главным недостатком системы является качество получаемого реферата: отобранные предложения не полностью передают смысл текста.

Таким образом, у инструментов, использующих более широкий спектр методов реферирования, нет поддержки русского языка, а инструменты, в которых имеется возможность реферирования текста на русском языке, используют только статистический или статистический и позиционный методы. Следовательно, существует необходимость в создании инструментов с возможностью реферирования текстов на русском языке, использующих более широкий набор методов реферирования.

Сравнительный анализ методов автоматического реферирования

Определение качества работы методов реферирования также является отдельным направлением исследования, так как отсутствуют какие-либо наборы эталонных рефератов. В ряде исследовательских работ [2] упоминаются подготовленные или полученные корпусы текстов новостей с их рефератами, но они не находятся в свободном доступе и содержат перефразированные предложения, передающие смысл исходного текста, что делает невозможным их применение для оценки качества реферирования методом экстракции.

Таким образом, для получения результатов работы описанных методов был подготовлен набор текстов на русском языке разных стилей — проведен опрос респондентов в целях получения эталонных рефератов. Эталонные рефераты — упорядоченные списки предложений по важности их в тексте на основании усредненной оценки группы респондентов разного возраста, профессии и т. д.

Были реализованы описанные методы автоматического реферирования: статистический, позиционный, индикаторный. Статистический метод реализован с применением статистического и симметричного алгоритмов расчета веса предложений. Индикаторный метод реализован с применением алгоритма на основе использования только индикаторов и алгоритма на основе использования маркеров, индикаторов и коннекторов.

В табл. 1 приведены результаты работы методов автоматического реферирования текстов в зависимости от стилей текста и объема реферата 30 %, 50 % и 70 %. В табл. 1 представлены результаты (в процентах — отношение предложений реферата полученного каким-либо методом к "эталонному") для двух разноплановых (хорошо структурированный и более описательный) текстов публицистического стиля, поскольку публицистическому стилю характерно применение разных способов изложения текста, развития мысли и т. д.

Анализ полученных результатов показал, что:

1) наиболее эффективный метод — статистический с применением симметричного алгоритма, в нем выбираются связанные друг с другом предложения, что позволяет достичь смысловой целостности реферата;

2) наименее эффективный метод — индикаторный (только индикаторы), так как для построения реферата недостаточно только находящихся в тексте индикаторов;

3) недостаток позиционного метода — отсутствие возможности получения реферата заданного объема;

4) недостаток индикаторного метода — большая зависимость от пользователя. Отсутствие маркеров и коннекторов уменьшает эффективность работы метода на 2,3—25,4 %;

Таблица 1

Результаты работы методов автоматического реферирования текстов

Стиль текста	Объем реферата, %	Статистический метод с применением статистического алгоритма	Статистический метод с применением симметричного алгоритма	Позиционный метод	Индикаторный метод	Индикаторный метод (только индикаторы)	Среднее значение
Публицистический	30	20	40		40	80	**46**
	50	45	56	50	78	67	**59,2**
	70	77	69		85	69	**70**
Среднее значение		**47,3**	**55**	**50**	**67,7**	**72**	
Публицистический	30	67	67		67	33	**59,6**
	50	64	64	64	64	55	**62,2**
	70	80	87		80	80	**78,2**
Среднее значение		**70,3**	**72,7**	**64**	**70,3**	**56**	
Научный	30	20	40		40	20	**43**
	50	63	63	75	63	25	**57,8**
	70	82	82		82	64	**77**
Среднее значение		**55**	**61,7**	**75**	**61,7**	**36,3**	
Художественный	30	42	42		42	33	**40,2**
	50	65	75	42	55	40	**55,4**
	70	71	82		71	64	**66**
Среднее значение		**59,3**	**66,3**	**42**	**56**	**45,7**	

5) с увеличением объема реферата увеличивается процент совпадений, что обусловлено увеличением числа ключевых слов и более полной передачей смысла текста;

6) наиболее эффективные результаты получаются при реферировании научных и публицистических текстов, а наименее — при реферировании художественных, так как для художественных текстов характерна неоднозначность их понимания, что отрицательно влияет на работу методов автоматического реферирования [7].

Алгоритмы на основе сочетания методов автоматического реферирования текстов

Проведенный сравнительный анализ работы методов автоматического реферирования текста показал, что каждый из них имеет определенные недостатки, многие из которых могут быть компенсированы дополнительным применением другого метода. Исходя из этого были разработаны алгоритмы на основе сочетания этих методов.

- *Алгоритм на основе сочетания позиционного метода со статистическим методом*

Недостатком позиционного метода является отсутствие возможности получения реферата заданного объема. При применении позиционного метода весь текст делится на ключевые предложения — те, которые находятся в начале и конце каждого абзаца, и не ключевые — остальные предложения текста. Все ключевые предложения имеют одинаковую значимость. Следовательно, становится невозможным увеличивать или уменьшать объем реферата.

Благодаря совместному использованию позиционного и статистического метода появляется критерий для оценки значимости предложения — его вес. Это значит, что объем реферата может быть изменен в зависимости от полученных весов предложений.

- *Алгоритм на основе сочетания индикаторного метода со статистическим методом*

Результаты работы индикаторного метода в большой степени зависят от человека, что является его основным недостатком этого метода. Реферат строится в зависимости от встречающихся в тексте маркеров, индикаторов и коннекторов, причем маркеры и коннекторы определяются человеком. Таким образом, если человек, применяющий этот метод, проигнорирует данную возможность или некорректно задаст маркеры и коннекторы, то вес предложений, который будет определяться только на основе находящихся в них индикаторов, будет сформирован неверно, что от-

рицательно скажется на результатах работы метода.

Применение сочетания индикаторного метода со статистическим методом позволяет использовать автоматически выделенные ключевые слова. Вес предложения в этом случае будет вычисляться на основе ключевых слов, выделяемых при использовании статистического метода, и индикаторов, используемых индикаторным методом.

Так как маркеры — ключевые слова и коннекторы — синонимы определяются человеком, то неправильное их выделение приведет к ухудшению работы методов. В результате опроса респондентов было выявлено, что задача выделения синонимов является непростой для большинства из них. Многие респонденты выделяют слова, основываясь на своих ассоциациях, которые при этом означают разные предметы, явления, действия, т. е. не являются синонимами, а находятся в отношении "род—вид" или связаны только ассоциативно. В результате реферат на основе индикаторного метода и алгоритма сочетания его со статистическим методом строится с использованием "ошибочных" синонимов, что приводит к ухудшению их работы. Таким образом, при реализации инструментов реферирования ввод синонимов человеком был исключен, впоследствии он может быть заменен использованием тематических словарей синонимов в сочетании с автоматическим определением предметной области текста [8, 9].

Интегральный метод автоматического реферирования текстов

На основе проведенного анализа полученных результатов применения алгоритмов реферирования текстов на основе сочетания различных методов был разработан интегральный метод автоматического реферирования текста.

Интегральный метод автоматического реферирования текста основан на комплексном использовании следующих методов:

- статистического с применением статистического алгоритма;
- статистического с применением симметричного алгоритма;
- позиционного;
- позиционного-статистического с применением статистического алгоритма;
- позиционного-статистического с применением симметричного алгоритма;
- индикаторного с использованием маркеров и индикаторов;
- индикаторного с использованием только индикаторов;
- индикаторного-статистического с применением статистического алгоритма;
- индикаторного-статистического с применением симметричного алгоритма.

Лежащий в основе интегрального метода алгоритм получения реферата состоит из следующих шагов:

1. Выделение ключевых предложений вышеописанными методами.

2. Расчет числа повторений каждого ключевого предложения:

$$k = \sum_{i=1}^{n} F_i(s), \qquad (1)$$

где s — ключевое предложение; $F_i(s)$ — функция, определяющая наличие ключевого предложения s во множестве выделенных i-м методом ключевых предложений; n — число методов;

$$F(s) = \begin{cases} 1, s \in M; \\ 0, s \notin M, \end{cases} \qquad (2)$$

где s — слово; M — множество методов выделений ключевых предложений.

3. Расчет порога пересечения:

- формирование числового ряда из полученных значений числа повторений всех ключевых предложений;
- определение моды полученного числового ряда p, p является порогом пересечения ключевых предложений.

4. Выделение предложений, число повторений которых больше или равно пороговому значению.

5. Сравнение числа полученных предложений с запрашиваемым объемом реферата:

- если число полученных предложений меньше, чем запрашиваемый объем реферата, то добавляются предложения, число повторений которых наиболее близко к порогу пересечения;
- если число полученных предложений больше, чем запрашиваемый объем реферата, то удаляются предложения с наименьшим числом повторений.

При получении ключевых предложений могут встречаться предложения с одинаковым числом повторений. В этом случае невозможно определить, какое из них следует удалить или добавить, поэтому используется одно, выбранное случайным образом предложение.

6. Сортировка полученных предложений по их номерам и получение итогового списка ключевых предложений и реферата.

В табл. 2 приведены результаты сравнения работы алгоритмов интегрального метода.

Анализ полученных результатов показал, что:

1) применение интегрального метода с использованием маркеров позволяет получить более эффективные результаты, чем без их использования, так как реферат строится на основе автоматически выделенных ключевых слов и введенных пользователем маркеров;

2) использование интегрального метода с применением алгоритма случайного выбора предложений с использованием маркеров позволяет получить более эффективные результаты, чем без их использования, так как реферат строится на основе автоматически выделенных ключевых слов и введенных пользователем маркеров;

3) использование интегрального метода с применением алгоритма случайного выбора предложений позволяет получать каждый раз разный реферат, что предоставляет пользователю возможность выбора наиболее подходящего из них;

4) в некоторых случаях использование интегрального метода с применением алгоритма случайного выбора предложений позволяет получить более эффективные результаты, чем использование "чистого" интегрального метода, так как выбираются предложения, число повторений которых меньше порогового значения, но которые являются ключевыми предложениями по мнению респондентов.

В табл. 3 приведены результаты сравнения работы интегрального метода и других существующих инструментов автоматического реферирования.

По данным табл. 3 можно сделать вывод, что использование интегрального метода в большинстве случаев позволяет получить более эффективные результаты (в среднем на 15,7 %), чем использование других инструментов реферирования.

На рис. 1 представлен график зависимости времени построения реферата от объема исходного текста. Предложенный интегральный алгоритм имеет линейную сложность, что позволяет использовать его в качестве одного из

Таблица 2

Результаты сравнения работы алгоритмов интегрального метода

Стиль текста	Объем реферата, %	Интегральный метод	Интегральный метод (случайный выбор)	Интегральный метод (без маркеров)	Интегральный метод (случайный выбор, без маркеров)
Публицистический	30	40	38,4	20	33,7
	50	67	56,4	56	50
	70	77	68,3	69	64,7
Среднее значение		61,3	54,4	48,3	51,5
Публицистический	30	67	58,8	67	47,8
	50	73	66,8	73	62,5
	70	87	74,4	87	69,8
Среднее значение		75,7	66,7	75,7	60,1
Научный	30	40	43,2	40	36,3
	50	75	61,7	63	53,7
	70	82	76,9	82	79,4
Среднее значение		65,7	60,6	61,7	56,5
Художественный	30	33	31,5	33	30,3
	50	60	63,3	55	58,7
	70	75	74,2	75	76,2
Среднее значение		56	56,3	54,3	**55,1**

Таблица 3

Результаты сравнения работы интегрального метода и других существующих инструментов автоматического реферирования

Стиль текста	Объем реферата, %	Интегральный метод	Text Summarization	Tools4noobs
Публицистический	30	40	40	60
	50	67	67	67
	70	77	77	69
Среднее значение		**61,3**	**61,3**	**65,3**
Публицистический	30	67	17	50
	50	73	45	55
	70	87	60	73
Среднее значение		**75,7**	**40,7**	**59,3**
Научный	30	40	20	20
	50	75	50	63
	70	82	64	82
Среднее значение		**65,7**	**44,7**	**55**
Художественный	30	33	25	17
	50	60	45	40
	70	75	71	61
Среднее значение		**56**	**47**	**39,3**

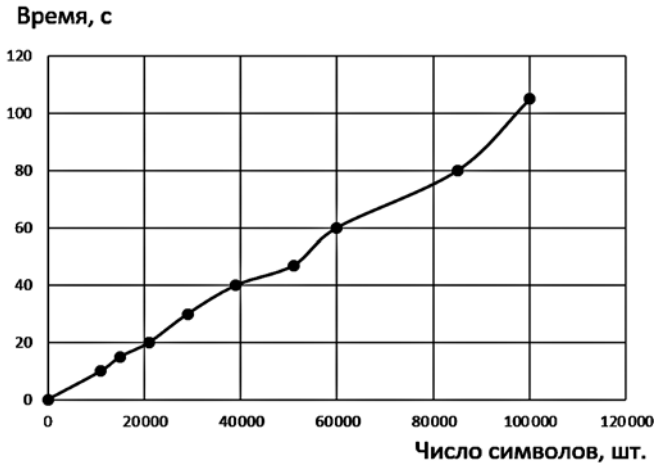


Рис. 1. График зависимости времени построения реферата от объема исходного текста

этапов при обработке больших наборов текстов, в том числе в прикладных системах.

Библиотека автоматического реферирования текста

С использованием всех реализованных методов была разработана Java-библиотека Summarization [10] для обеспечения возможности использования этих методов в других системах и программных комплексах.

Входящие в разработанную библиотеку методы делятся на две группы [10].

В первую группу входят методы, предназначенные для работы с текстом:

1. Метод для получения всех слов текста.
2. Метод для получения всех ключевых слов текста интегральным методом [11].
3. Метод для разделения текста на абзацы.
4. Метод для разделения абзацев на предложения.
5. Метод для получения слов каждого предложения в форме, в которой они употребляются в тексте.
6. Метод для получения слов каждого предложения в начальной форме.
7. Метод для получения ключевых слов каждого предложения.

Во вторую группу входят методы, предназначенные для получения реферата вышеописанными методами, и метод для получения списка предложений без их номеров в исходном тексте.

Сервис автоматического реферирования текста

С использованием разработанной библиотеки был создан веб-сервис с графическим интерфейсом пользователя и программным REST-интерфейсом, что позволяет использовать реализованные методы реферирования как пользователями, так и сторонними системами [10].

Сервис автоматического реферирования текста доступен на портале "Автоматизированный анализ текста" по адресу: http://abstracts.textanalysis.ru/, он предоставляет возможность получения реферата и ключевых слов. Для этого необходимо выполнить следующие шаги:

1. Вставить исходный текст в текстовое поле.
2. Выбрать метод реферирования. Все методы разделены на две группы в зависимости от того, необходимо ли наличие маркеров для применения метода (рис. 2).
3. Указать объем реферата в процентах.
4. Ввести маркеры, если выбран соответствующий метод.
5. Нажать кнопку "Получить реферат!".

В результате обработки ответа полученный реферат и ключевые слова отобразятся в соответствующих текстовых полях (рис. 3).

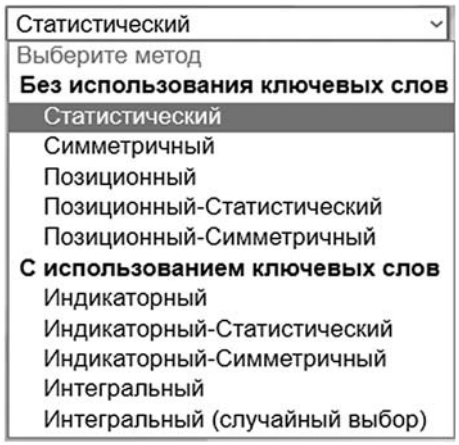


Рис. 2. Выпадающий список для выбора метода реферирования

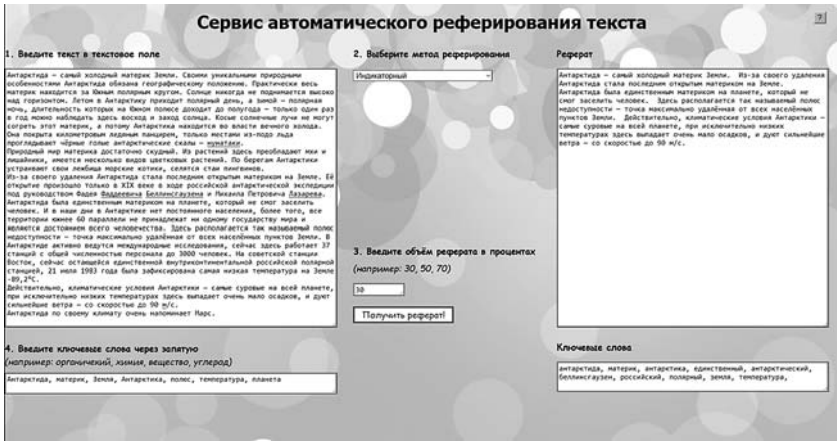


Рис. 3. Пример полученного реферата

Если пользователь не выбрал метод реферирования и дополнительные настройки, то будет автоматически применен интегральный метод реферирования текста.

Также пользователь может получить справочную информацию по использованию сервиса.

На основе полученных отзывов пользователей сформированы направления развития сервиса реферирования:

- определение числа символов, слов, предложений, абзацев по вводимым текстам и получаемым рефератам;
- создание возможности задания объема реферата не только в процентах, но в числе предложений и символов;
- создание возможности выбора предложений, которые должны быть включены в реферат;
- разбиение получаемого реферата на абзацы в соответствии с абзацами исходного текста;
- выделение ключевых слов в получаемом реферате.

Заключение

Разработанный интегральный алгоритм реферирования позволяет получать более точные по сравнению с эталонным рефератом результаты, чем другие существующие инструменты. Помимо развития сервиса реферирования дальнейшими направлениями исследования для совершенствования методов реферирования являются:

1. Использование ключевых словосочетаний и именованных сущностей в дополнение к ключевым словам при применении методов автоматического реферирования.

2. Дальнейшее исследование методов реферирования, в том числе дополнение интегрального метода распространенными алгоритмами для английского языка TextRank, LexRank, LSA.

3. Проведение дополнительного анализа характеристик текста (стиль, объем) с последующим автоматическим выбором наиболее подходящего метода.

4. Использование инструментов для устранения орфографических и пунктуационных ошибок, расшифровки сокращений в исходном тексте.

5. Использование подготовленных по разным тематикам и стилям текстов словарей синонимов [8] при применении индикаторного метода и алгоритмов его сочетания со статистическим методом.

6. Выделение определенного числа ключевых слов в зависимости от объема текста.

Библиотека и сервис автоматического реферирования текста предоставляют возможность использования не только существующих методов автоматического реферирования, а также алгоритмов на основе их сочетания и интегральный метод. С помощью разработанного сервиса были получены рефераты по текстам различных стилей и объемов и разным видам технической документации, программный интерфейс сервиса использован в мобильном приложении Tourist Helper 2.0 [12].

Сервис полезен для специалистов, работающих с большим объемом текстовых данных, и позволяет быстро получить краткий вариант текста нужного объема. Наличие программного интерфейса сервиса и библиотеки в свободном доступе дает возможность автоматизации сокращения текстовых данных в рамках любого процесса.

Список литературы

1. **Маркушевская Л. П., Цапаева Ю. А.** Аннотирование и реферирование: методические рекомендации для самостоятельной работы студентов. СПб.: ИТМО, 2008. 8 с.

2. **Браславский П. И., Колычев И. С.** Автоматическое реферирование веб-документов с учетом запроса // Интернет-Математика 2005. Автоматическая обработка веб-данных. М.: Яндекс, 2005. С. 485—501.

3. **Тарасов С. Д.** Современные методы автоматического реферирования // Научно-технические ведомости СПбГПУ. 2010. № 6. С. 59—74.

4. **Букатникова С. Д.** Понимание текста как проблема современной лингвистики и гуманитарного познания // Молодой ученый. 2015. № 6. С. 799—803.

5. **Лазарева О. Ю., Боломутова М. С.** Методы выделения ключевых слов в контексте электронных обучающих систем // Молодой ученый. 2016. № 22. С. 143—146.

6. **Яцко В. А.** Симметричное взвешивание терминов // Символ науки. 2015. № 12. С. 87—88.

7. **Позняк Г. В.** Психолингвистические основы реферирования как учебной деятельности // Факультет международных отношений БГУ: электронный сборник. Вып. II. Минск: БГУ, 2012. С. 33—38.

8. **Милованова Е. Е.** Определение семантической схожести текстов информационными системами // Гагаринские чтения. 2019: XLV Международная молодежная научная конференция: Сборник тезисов докладов. М.: Московский авиационный институт (национальный исследовательский университет), 2019. С. 381.

9. **Белов С. М.** Создание программной системы классификации текстов // Материалы XVIII Международной конференции "Информатика: проблемы, методология, технологии". Секция 10: компьютерная лингвистика. 2018. Т. 6. С. 8—12.

10. **Документация** Java-библиотеки Summarization и сервиса автоматического реферирования текста. URL: http://textanalysis.ru/jce/details/our-instr, свободный. (Дата обращения: 14.06.19).

11. **Иващенко М. В.** Анализ методов автоматизированного выделения ключевых слов из текстов на естественном языке // Материалы XVIII Международной конференции "Информатика: проблемы, методология, технологии". Секция 10: компьютерная лингвистика. 2018. Т.6. С. 19—24.

12. **Приложение** Tourist Helper 2.0 / Google Play. URL: https://play.google.com/store/apps/details?id=ru.textanalysis.touristhelper, свободный (дата обращения: 20.06.19).

E. V. Politsyna, Cand. of Tech. Sc., Associate Professor, e-mail: kathrin.beaver@mail.ru,
S. A. Politsyn, Cand. of Tech. Sc., Associate Professor, e-mail: pul_forever@mail.ru,
A. O. Kasatkina, Student, e-mail: alyona.kasatkina1997@yandex.ru,
Moscow Aviation Institute (National Research University), Moscow, Russian Federation

Development of Integrated Algorithm and Tools of Automatic Summarization for Texts in the Russian Language

The amount of information provided in text form is increasing every year. Thus, the task of automatic natural language text processing (NLP), and especially the reduction of its volume, is becoming increasingly important. But tools which produce abstracts in the Russian language mainly use the statistical method only. So, there is a need to research the summarization algorithms and create new algorithms and tools that will use several automatic summarization methods simultaneously to improve the results. The article shows the results of extraction algorithms, basing of which an integrated summarization algorithm, a library and service of automatic text summarization in the Russian languages are proposed to enable the use of various reference methods.

Keywords: *data extraction, methods of summarization, integrated summarization method, automatic summarizations service*

DOI: 10.17587/it.26.30-38

References

1. **Markushevskaya L. P., Tsapaeva Y. A.** Annotation and summarization: guidelines for students' self-studies, Spb., Publishing house of ITMO, 2008, 8 p. (in Russian).

2. **Braslavsky P. I., Kolychev I. S.** Automated summatization of web-ducuments basing on search requests, In proceedings: Internet-matematika 2005. Automated web-data processing. Moscow, Yandex, pp. 485—501 (in Russian).

3. **Tarasov S. D.** Modern methods of automated summarization, *Nauchno-technicheskie vedomosti SpbGpu*, 2010, no. 6, pp. 59—74 (in Russian).

4. **Bukatnikova S. D.** Text understanding as a modern problem of linguistics and humanitarian knowledge, *Molodoy Ucheniy*, 2015, no. 6, pp. 799—803 (in Russian).

5. **Lazareva O. Y., Bolotoumova M. S.** Methods of keyword extraction in computer educational systems, *Molodoy Ucheniy*, 2016, no. 22, pp. 143—146 (in Russian).

6. **Yatsko V. A.** Symmetric term weighting, Simvol nauki, 2015, no. 12, pp. 87—88 (in Russian).

7. **Posnyak G. V.** Psycholinguistic basics of summarizing as a learning, *Faculty of International Affairs BGU:* vol. II, Minsk, BGU, 2012, pp. 33—38 (in Russian).

8. **Milovanova E. E.** Determination of semantic similarity of texts, *XLV Mezhdunarodnaya molodezhnaya nauchnaya konferenciya: sbornik dokladov*, Moscow, MAI, 2019, p. 381 (in Russian).

9. **Belov S. M.** Development of the text classification system, *Materialy XVIII Materials of XVIII International Conference "Informatika: problemy, metodologiya, tekhnologii",* vol. 6, pp. 8—12 (in Russian).

10. **Summarization** library Java-docs and automated summarization service documentation, available at: http://textanalysis.ru/jce/details/our-instr (access date: 14.06.19).

11. **Ivashchenko M. V.** Analysis of automated keyword extraction methods in natural language texts, *Materials of XVIII International Conference "Informatika: problemy, metodologiya, tekhnologii". Sec. 10: computer linguistics*, 2018, vol. 6, pp. 19—24 (in Russian).

12. **Application** Tourist Helper 2.0, Google Play, available at: https://play.google.com/store/apps/details?id=ru.textanalysis.touristhelper. — (access date: 20.06.19).

УДК 519.25, 004.93 + 621.397.13.037.372 DOI: 10.17587/it.25.39-45

Я. А. Хасан[1], аспирант, e-mail: midocom@mail.ru,
Н. Г. Рыжов[1], канд. техн. наук, доц., e-mail: ngryzhov@mail.ru,
Ш. С. Фахми[1,2], д-р техн. наук, доц., e-mail: shakeebf@mail.ru,
Е. В. Костикова[3], канд. техн. наук, доц., e-mail: kostikovaev@mail.ru,
[1]Санкт-Петербургский государственный электротехнический университет "ЛЭТИ",
[2]Институт проблем транспорта им. Н. С. Соломенко РАН,
[3]Государственный университет морского и речного флота имени адмирала С. О. Макарова

Адаптивный способ спектрального преобразования видеоинформации транспортных изображений

Предложен метод кодирования и декодирования видеоинформации на основе адаптивного трехмерного дискретного косинусного преобразования, обеспечивающий повышение эффективности устройств передачи и уменьшение информационных показателей качества видеосистем: уровни искажения, скорости передачи и сложности кодирующих устройств. Для повышения производительности алгоритма сжатия изображений используется классификация транспортных сюжетов по типу движения. Получены результаты тестирования алгоритма и приведен сравнительный анализ предложенного метода с известными методами MPEG2 и MPEG4.

Ключевые слова: *трехмерное косинусное преобразование, видеоинформации, адаптивность, сложность, сканирование*

Введение

Методы кодирования и декодирования изображений за последние десятилетия стали весьма актуальной областью исследований [1]. Главной задачей обработки видеоинформации является сокращение избыточности на изображениях и построение компактных форм представления изображений для хранения и передачи визуальных данных по каналу связи [2]. Важная объединяющая характеристика большинства изображений заключается в том, что имеются локальные области, где соседние пиксели сильно коррелированны и, следовательно, имеют избыточную информацию. Для более компактного и экономного представления изображения используются различные схемы декорреляции, которые разделяются на два класса: пространственные [3] и трансформационные (спектральные) [4].

Суть трансформационного кодирования изображений часто заключается в применении дискретного косинусного преобразования (ДКП) [5, 6], обеспечивающего хороший компромисс между способностью сжатия видеоинформации и вычислительной сложностью. Наиболее распространенным стандартом является JPEG [7] — известная технология сжатия, использующая ДКП и достигающая высокой степени сжатия с меньшей потерей данных. Другим стандартом сжатия видео является MPEG [8], обеспечивающий лучший коэффициент сжатия видеоизображений с низкой скоростью передачи при высоком качестве восстановления.

В данной статье предлагается новый адаптивный метод кодирования и декодирования изображений на основе адаптивного трехмерного ДКП (АДКП-3D) с использованием модифицированной таблицы квантования для кубов различного размера [9].

В работе предложена методика оценки информационных показателей качества устройств сжатия изображений с использованием ДКП-3D. Предложенная методика использует модифицированную таблицу квантования и метод преобразования куба трехмерного изображения в одномерный массив (1D), обеспечивая лучшую эффективность кодирования путем изменения размера куба для ДКП с уче-

том пространственной корреляции в пределах одного кадра и временной межкадровой корреляции по времени. Для повышения производительности алгоритма сжатия изображений используется пирамидально-рекурсивный метод [10, 11] для определения неравномерной сетки, которая в дальнейшем и определяет размер кубов в последовательности кадров видеопотока. Результирующее 2D-изображение преобразуется в 3D-кубики, размеры которых определяются на основе типа изображения (низкая или высокая детализация). Затем ДКП-3D применяется к полученным кубам.

Параллельная реализация предложенного алгоритма используется для увеличения времени вычислений двумя способами: первый использует процесс распараллеливания вычислений с использованием SPMD (одна программа с несколькими данными), а другой метод использует графический процессор (GPU) с программно-аппаратной архитектурой параллельных вычислений (CUDA), которая позволяет существенно увеличить вычислительную производительность.

Производительность предложенного алгоритма оценивается с использованием некоторых наиболее часто используемых в литературе по сжатию тестовых изображений. Результаты тестирования показывают, что предложенный алгоритм превосходит несколько методов сжатия по соотношению сигнал/шум (PSNR) и скорости сжатия.

Постановка задачи исследования

Оценка движения играет ключевую роль в каждом видеокодере. Например, в гибридном кодировании с компенсацией движения в известных кодеках, таких как кодеки семейства MPEG и H.26x. При этом движение напрямую влияет на временную точность прогнозирования и является основным фактором эффективности кодера. В предложенном методе кодирования на основе адаптивного косинусного преобразования движение также играет решающую роль, поскольку оно напрямую влияет на способ определения размера кубов и соответствующего коэффициента сканирования. Однако в АДКП-3D-кодировании движение может быть представлено явно в пространственно-временной области в виде вектора или неявно через параметры ориентации расположения доминантной энергии в пространственно-временном частотном пространстве. Эта двойственность представления движения позволяет разработать два различных подхода к оценке вектора параметров ориентации, необходимых для оптимального размера кубов и способа сканирования коэффициентов ДКП.

Появление в последнее время технологии "система-на-кристалле" и современных САПР на базе репрограммируемых логических интегральных схемах привело к формализованному учету сложности устройств кодирования и декодирования изображений. Кроме того, такая формализация была необходима для создания и развития новой элементной базы вычислительной техники в виде сложно-функциональных блоков в составе видеосистем-на-кристалле [12, 13].

Оптимизация алгоритмов кодирования и декодирования видеоинформации должна выполняться с учетом не только точности восстановления (среднеквадратического отклонения, СКО) и скорости создания кода (R — числа бит на пиксель), как принято в большинстве реализуемых на сегодняшней день алгоритмах, но и с учетом сложности устройств (W) [13]. Эти три величины взаимосвязаны, и выбор оптимальных параметров устройств кодирования и декодирования необходимо осуществлять по критерию минимума целевой функции (P), включающей весовые коэффициенты $\{c_i\}$ этих трех информационных показателей качества: степени искажения восстановленных изображений (ΔI), скорости передачи (R) и сложности устройств кодирования (W) [14].

Таким образом, задача оптимизации при синтезе видеосистем-на-кристалле в целом сводится к поиску оптимальных значений информационных показателей качества при заданных соответствующих весовых коэффициентах, в то же время обеспечивающих минимум целевой функции [15]:

$$P = c_0\Delta I(\varepsilon) + c_1 R(\varepsilon) + c_2 W_{\text{к}}(\varepsilon) + c_3 W_{\text{д}}(\varepsilon) \to \min.$$

Заметим, что определение множества решений с учетом ограничений для данной функции является труднейшим этапом всего процесса оптимизации при проектировании видеосистем из-за антагонистических и конфликтных ситуаций, в условиях вероятностной и нечеткой неопределенности статистических характеристик нестационарного источника изображений.

В данной работе решаются задачи:

- разработки адаптивного метода спектрального косинусного преобразования изобра-

жений, учитывающего пространственно-временную корреляцию видеопотоков, полученных из различных камер наблюдения на транспорте;
- определения информационных показателей качества системы кодирования изображений: СКО и скорости передачи (R);
- сравнения результатов исследований с известными методами кодирования и декодирования изображений (MPEG2 и MPEG4) для различных классов изображений транспортной системы наблюдения.

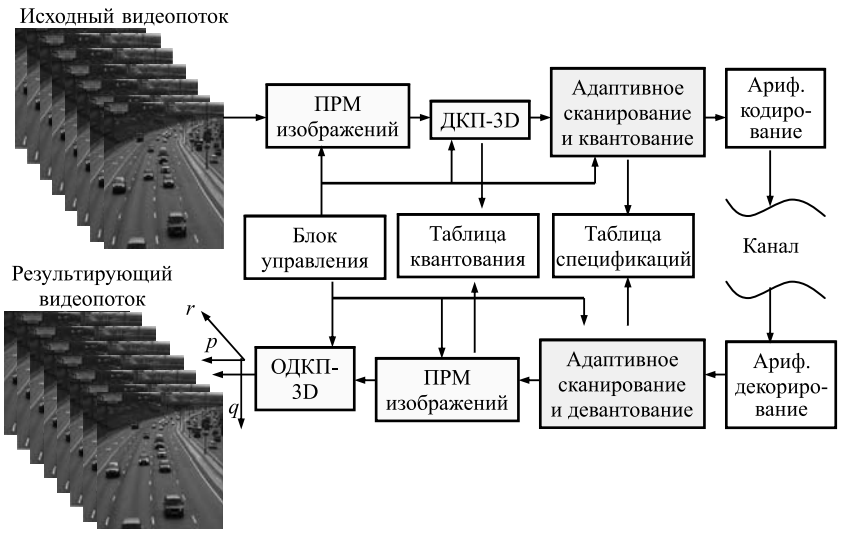


Рис. 1. Структурная схема устройства видеосистемы передачи изображений транспортных сюжетов

Описание предложенного адаптивного метода ДКП-3D

Адаптивное трехмерное ДКП (АДКП-3D). ДКП представляет собой важный инструмент декорреляции из-за симметричности функции косинуса и осуществляет уплотнение энергии путем извлечения только необходимых коэффициентов частотной области сигнала. Другими словами, ДКП позволяет четко разделить исходное изображение на две области — высокочастотную и низкочастотную [1, 4].

Процесс кодирования и декодирования изображений включает четыре этапа (рис. 1).

Этап 1. Формирование неравномерной сетки ДКП

Для определения оптимальных размеров кубов, подвергающихся в дальнейшем АДКП-3D, проводится анализ пространственной корреляции локальных областей первого кадра куба и временной корреляции для определения числа кадров куба. Анализ пространственной корреляции и разбиение изображения на блоки различной формы и размера осуществляется на основе пространственно-рекурсивного метода (ПРМ) разбиения [10, 11, 13].

Анализ временной корреляции выполняется на основе сравнения спектральных коэффициентов ДКП кубов размера 4×4×4 в пределах кубов большего размера (32×32×32 или 64×64×64).

Для определения типа движения и соответствующего числа кадров в кубе вычисляется разность (Δ) коэффициентов первого кадра и последнего и сравнивается с заданными порогами (ρ_1 и ρ_2):

- при $\Delta \leqslant \rho_1$ — движение отсутствует, и ДКП по времени не выполняется;
- при $\rho_1 \leqslant \Delta \leqslant \rho_2$ — движение слабое, и размер куба устанавливается равным 16×16×16;
- при $\rho_2 \leqslant \Delta \leqslant \rho_3$ — движение среднее, и размер куба равен 8×8×8;
- при $\Delta \geqslant \rho_3$ — движение сильное, и размер куба равен 4×4×4.

Этап 2. ДКП-3D

Для заданной трехмерной последовательности пространственных данных $\{X_{ijk}; i, j, k = 0, 1, ..., N-1\}$ трехмерная последовательность данных ДКП-3D$\{Y_{pqr}; p, q, r = 0, 1, ..., N-1\}$ определяется по формуле:

а) для прямого АДКП-3D:

$$Y_{pqr} = E_p E_q E_r \sqrt{\frac{z}{N^3}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} X_{ijk} \times$$

$$\times \frac{\cos(2i+1)p\pi}{2N} \cdot \frac{\cos(2j+1)q\pi}{2N} \cdot \frac{\cos(2k+1)r\pi}{2N},$$

где $Ex = \begin{cases} \sqrt{1/\sqrt{2}}, \text{ при } x = 0, \\ 1, \text{ при } x \neq 0, \end{cases}$ где $x = p, q, r$;

б) для обратного ДКП-3D (ОДКП-3D):

$$X_{ijk} = \sqrt{\frac{z}{N^3}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} Y_{pqr} \times$$

$$\times \frac{\cos(2i+1)p\pi}{2N} \cdot \frac{\cos(2j+1)q\pi}{2N} \cdot \frac{\cos(2k+1)r\pi}{2N},$$

где значение z определяется размером куба.

Параллельная архитектура для вычисления АДКП-2D основана на методе декомпозиции фреймов строк и столбцов и использует один и тот же одномерный АДКП-1D модуль для всех трех измерений.

Этап 3. Адаптивное сканирование и квантование

С этапом квантования связано понятие визуальной избыточности. Важно на этом этапе, с одной стороны, учитывать взаимосвязь анизотропии яркостей и способа сканирования для достижения нужного коэффициента сжатия, а с другой стороны, сохранять качество при заданном уровне искажений. В работе в качестве исходных визуальных данных были взяты видеопотоки из различных камер наблюдения на дорогах мегаполиса. Вся видеоинформация транспортной специфики разделена на три группы в зависимости от типа движения и интенсивности транспортных средств (рис. 2).

В результате исследования большого объема видеоданных были сформулированы три класса видеопотоков: 1) видеопоток без движения; б) видеопоток со средним движением; в) видеопоток с высоким движением.

Адаптивное квантование выполнялось с порогами, полученными с учетом статистических характеристик распределения спектральных коэффициентов и уровнями — в серединах интервалов квантования [16, 17]. По форме пространственного распределения спектральных коэффициентов ДКП и по оценке степени корреляции использованы следующие способы сканирования (рис. 2): горизонтальные, вертикальное и зигзагообразное. Если в пределах блока яркости пикселей не превышают заданного минимального порога, то ДКП не выполняется, и на этапе восстановления всем пикселям присваивается среднее значение яркости.

Этап 4. Арифметическое кодирование

Завершающим этапом процесса кодирования является энтропийное кодирование. Подобно сжатию неподвижных изображений [18] используется кодирование Хаффмана. После квантования и адаптивного сканирования все ненулевые коэффициенты подвергаются кодированию длин серии рядом с предшествующими нулевыми коэффициентами, т. е. используется кодирование длины пробега [19]. Для 3D-коэффициентов ДКП были получены новые словари кодов (см. таблицу).

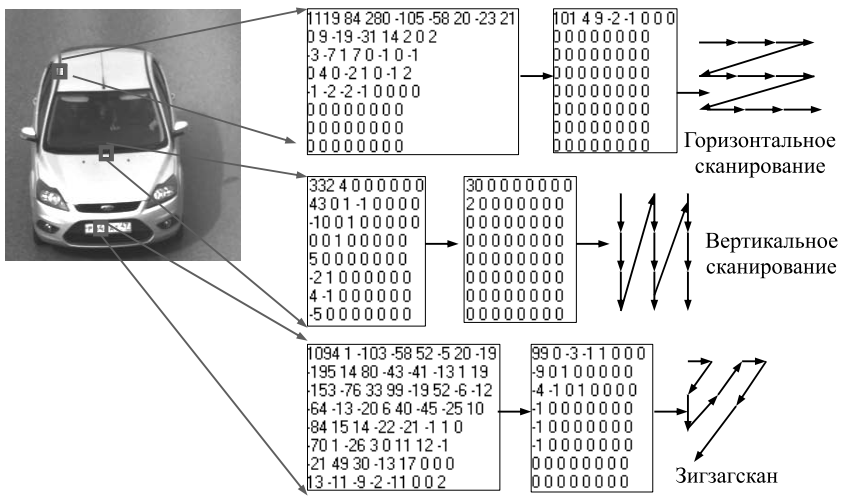


Рис. 2. Способы сканирования блоков ДКП изображений

Кодовые слова ДКП-3D

Значения		Коды		Длина
<0	>0	<0	>0	
0				0
−1	1	0	1	1
−1 −3	2,3	00,01	10,11	2
−7, −6, −5, −4	4,5,6,7	000,001,010,011	100,101,110,111	3
−15, ..., −8	8, ..., 15	0000, ..., 0111	1000, ..., 1111	4
−31, ..., −8	32, ..., 15	00000, ..., 01111	10000, ..., 11111	5
⋮	⋮	⋮	⋮	⋮
−2047, ..., 1024	1024, ..., 2047	0000000000,...	...,11111111111	11

Коэффициент в верхнем левом углу матрицы коэффициентов ДКП называется DC-коэффициентом (direct current coefficient), а все остальные коэффициенты — AC-коэффициентами (alternating current coefficient).

Как упоминалось выше, в пределах одного блока (куба) коэффициенты переменного тока переупорядочиваются с помощью модифицированного адаптивного сканирования (рис. 2), затем выполняется кодирование длин серий, которое заканчивается идентификатором EOB (End-Of-Block), обозначающим конец блока.

Динамический диапазон коэффициентов ДКП превышает в 8 раз динамический диапазон значений пикселей исходного изображения. Например, при 8-битном представлении отсчетов изображения его динамический диапазон составляет 256 дискретных уровней (значения от 0 до 255), а динамический диапазон коэффициентов спектра ДКП — 2048 значений уровней (от 0 до 2048 для DC-коэффициентов и от −1023 до +1024 для AC-коэффициентов).

Кодирование коэффициентов ДКП в таком широком динамическом интервале потребует в последующих блоках MPEG-кодера перехода от 8-битного к 11-битному коду. Для предотвращения усложнения кодера после ДКП проводится сжатие динамического диапазона сигналов коэффициентов ДКП за счет увеличения шага квантования в 8 раз. Эта операция сводится к делению полученных в матрице значений коэффициентов ДКП на 8. Результат деления затем округляется до ближайших целых значений уровней новой шкалы квантования. Например, если исходное значение коэффициента ДКП было 22, то после деления на 8 (22/8 = 2,75) и округления до ближайшего целого значения новое значение будет равно 3. При этом новый динамический интервал составит 256 дискретных уровней от −128 до +127. Если размер куба для ДКП $N = 8$, следовательно, куб содержит от 0 до 255 AC-коэффициентов. Подобно DC-коэффициентам, весь интервал возможных значений делится на 11 подынтервалов (так как максимальное значение ДКП-коэффициента = 2047), и теоретически их значения варьируются от 0 до 2047 (см. таблицу).

Иными словами, положительные значения прямо кодируются их двоичным представлением, а отрицательные — так же, но с заменой ведущей 1 на 0.

Согласно обширному словарю кодов, менее распространенные значения AC представлены широкими кодовыми словами (см. таблицу). Например, кодовое слово для субынтервала [8...15] имеет ширину 8 бит, для субынтервала [−31...15] требуется 10 бит и для субынтервала [−2047...2047] необходимо 22 бит.

Результаты исследований различных транспортных видеопотоков

Эксперименты проводились над тремя различными транспортными видеопотоками: с малым движением; со средним движением; с высоким движением.

Основными преимуществами использования косинусного преобразования с переменным размером кубов как в пространстве сигнала, так и по времени являются следующие:

а) незначительное уменьшение СКО по сравнению с известными ДКП-3D и MPEGx. При этом скорость передачи видеоинформации почти не различается для всех кодеков и равна $R = 0{,}01$ бит/пиксель (рис. 3), а коэффициент сжатия предложенного метода в 2...3 раза больше для случаев видеопотока со слабым и средним движением;

б) существенное снижение вычислительной сложности (в среднем в 4...5 раз) по сравнению с стандартом MPEG и увеличение в среднем в 1,5...2 раза по сравнению с известным косинусным преобразованием фиксированного размера блоков.

Потенциальным применением предлагаемого трехмерного адаптивного ДКП подхода

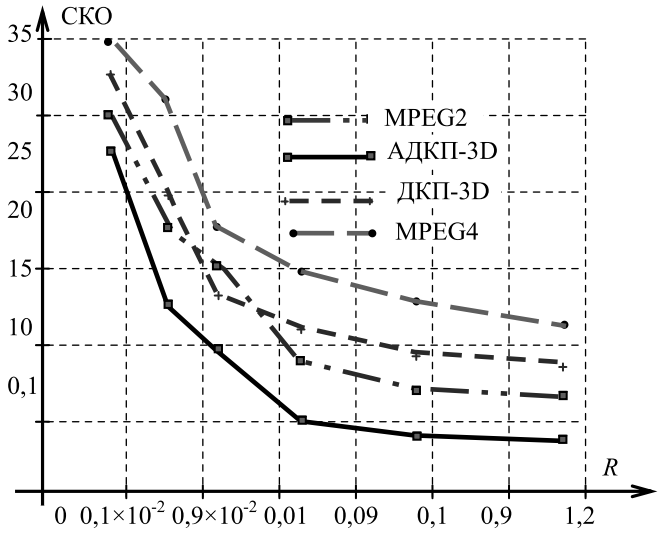


Рис. 3. Зависимость СКО от скорости передачи

могут быть портативные цифровые устройства, такие как мобильный телефон, беспилотники, интеллектуальные камеры наблюдения и др.

Заключение

Сжатие изображений является чрезвычайно важной частью современных интеллектуальных транспортных видеосистем наблюдения. Имея возможность сжимать видеоинформации до доли их исходного размера, можно сэкономить ценное и дорогое дисковое пространство памяти. Кроме того, с появлением технологии "система-на-кристалле", передача изображений с мест чрезвычайных ситуации и аварий в диспетчерский центр управления в реальном времени становится возможным в современных вычислительных системах.

Из вышеприведенного исследования можно сделать следующие основные выводы:

1) предлагаемый метод на основе адаптивного косинусного преобразования принимает меньшее значение среднеквадратического отклонения, чем методы MPEG2, MPEG4 и классический метод косинусного преобразования без адаптации размеров кубов в 2...2,5 раза;

2) применение адаптивного способа определения размеров кубов для косинусного преобразования позволило увеличить коэффициент сжатия на 10...20 % при сохранении субъективного качества по сравнению с вышеуказанными методами;

3) полученные статистические характеристики и зависимости числа блоков различного размера, подвергающихся косинусному преобразованию, от степени однородности областей изображений позволяют уменьшить вычислительную сложность устройств кодирования и декодирования видеоинформации в транспортных системах наблюдения в 2...3 раза.

Список литературы

1. **Gonzalez R. C., Woods R. E.** Digital Image Processing. Upper Saddle River, N.J: Prentice Hall, 2008. P. 547—635.

2. **Sayood K.** Introduction to Data Compression. Amsterdam, Boston: Elsevier, 2006. P. 6—10.

3. **Крюкова М. С., Фахми Ш. С.** и др. Методы, алгоритмы кодирования и классификация изображений морских судов // Морские интеллектуальные технологии. 2019. № 1 (43). Т. 3. С. 145—155.

4. **Ричардсон Ян.** Видеокодирование. H.264 и MPEG-4 — стандарты нового поколения. М.: Техносфера, 2005. 368 с.

5. **Фахми Ш. С., Зубакин И. А.** Адаптивный алгоритм кодирования видеоинформации на основе трехмерного дискретного косинусного преобразования // Изв. вузов России. Сер. Радиоэлектроника. 2010. Вып. 1. С. 49—54.

6. **Патент РФ № 2375838.** Способ кодирования и декодирования видеоинформации на основе трехмерного дискретного косинусного преобразования. Фахми Ш. С., Ибатуллин С. М. и др. Опуб. 10.12. 2009, Б. И. № 34.

7. **Wallace G. K.** The JPEG still picture compression standard // IEEE Transactions on Consumer Electronics. 1992. Vol. 38, N. 1. P. 18—34.

8. **Sikora T.** MPEG digital video-coding standards // IEEE Signal Processing Magazine. 997. Vol. 14, N. 5. P. 82—100.

9. **Mulla A., Baviskar J., Baviskar A., Warty C.** Image compression scheme based on zig-zag 3D-DCT and LDPC coding // Proc. International Conference on Advances in Computing, Communications and Informatics. 2014. P. 2380—2384.

10. **Фахми Ш. С., Альмахрук М. М., Соколов Ю. М., Бобровский А. И., Еид М. М., Салем А.** Точность, скорость и сложность устройств кодирования изображений по опорным точкам // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16, № 4. С. 678—689.

11. **Фахми Ш. С., Альмахрук М., Салем А., Бобровский А. И., Еи М. М.** Информационные показатели качества устройств кодирования по опорным точкам // Вопросы радиоэлектроники. Сер. Техника телевидения. 2016. Вып. 3. С. 86—91.

12. **Фахми Ш. С., Соколов Ю. М.** Видеосистемы на кристалле: новые решения в задачах обработки видео // XXIV межд. науч. метод. конф. "Современное образование: содержание, технологии, качество "Современное образование: содержание, технологии, качество". СПб., СПбГЭТУ "ЛЭТИ", 21 апреля. 2018. Т. 2. С. 17—19.

13. **Фахми Ш. С.** Концепция проектирования интеллектуальных транспортных видеосистем на основе технологии "система на кристалле" // Журнал университета водных коммуникации. 2013. Вып. II (XVIII). С. 79—88.

14. **Моисеев Н. Н.** Математические задачи системного анализа. М.: Наука, 1981. 488 с.

15. **Фахми Ш. С., Цыцулин А. К., Колесников Е. И., Очкур С. В.** Функционал взаимообмена сложности и точности систем кодирования непрерывного сигнала // Информационные технологии. 2011. № 4. С.71—77.

16. **Lee M. C., Chan K. W., Adjeroh D. A.** Quantization of 3D-DCT coefficients and Scan Order for Video Compression // Journal of visual communication and image representation. 1997. Vol. 8, N. 4. P. 405—422.

17. **Bozinovic N., Konrad J.** Scan order and quantization for 3D-DCT coding // S&T/SPIE Symposium on Image and Video Communications and Proc. Jul. 8—11, 2003.

18. **Mehanna A., Aggoun A., Abdulfatah O., Swash M. R., Tsekleves E.** Adaptive 3D-DCT based compression algorithms for integral images // Broadband Multimedia Systems and Broadcasting. 2013. P. 1—5.

19. **Priyadarshini K. S., Sharvani G. S.** A survey on parallel computing of image compression algorithms // Proc. International Conference, Computational Systems for Health & Sustainability. April 2015. P. 178—183.

Y. A. Hasan[1], Postgraduate, e-mail: midocom@mail.ru,
N. G. Ryzhov[1], Assistant Professor, e-mail: ngryzhov@mail.ru,
Sh. S. Fahmi[1,2], Professor, e-mail: shakeebf@mail.ru,
E. V. Kostikova[3], Assistant Professor, e-mail: kostikovaev@mail.ru,
[1]Saint Petersburg Electrotechnical University "LETI"
[2]Solomenko Institute of Transport Problems of the Russian Academy of Sciences
[3]Admiral Makarov State University of Maritime and Inland Shipping

Adaptive Method of Spectral Transformation of Video Information of Transport Images

Proposed the method of coding and decoding video data based on adaptive three-dimensional discrete cosine transform (3D-DCT), provide increased efficiency of operation of the transfer device and reduction of information quality, systems: distortion levelsf, bit rate, and complexity encoders.To improve the performance of the image compression algorithm, the classification of transport subjects by type of motion is used. The results of testing the algorithm and a comparative analysis of the proposed method with the known methods MPEG2 and MPEG4 are obtained.

Keywords: *three-dimensional cosine transformation, video information, adaptability, complexity, scanning*

DOI: 10.17587/it.25.39-45

References

1. **Gonzalez R. C., Woods R. E.** Digital Image Processing, Upper Saddle River, N.J, Prentice Hall, 2008, pp. 547—635.

2. **Sayood K.** Introduction to Data Compression, Amsterdam, Boston, Elsevier, 2006, pp. 6—10.

3. **Kryukov M. S., Fahmy S. S.** et al. Methods, algorithms of coding and classification of images of sea vessels, *Marine Intelligent Technologies*, 2019, vol.3, no. 1 (43), pp. 145—155 (in Russian).

4. **Richardson Ian.** Video coding. H. 264 and MPEG-4 — new generation standards. Moscow, Technosphere, 2005, 368 p. (in Russian).

5. **Fahmi Sh. S., Zubakin I. A.** Adaptive algorithm of video information coding based on three-dimensional discrete cosine transformation, *Izv. University of Russia. Ser. Radionics*, 2010, iss. 1, pp. 49—54 (in Russian).

6. **Patent RF № 2375838.** Method of encoding and decoding video based on three-dimensional discrete cosine transformation, Fahmy sh. S., Ibatullin S. M. et al, Pub. 10.12, 2009, B. I. № 34 (in Russian).

7. **Wallace G. K.** The JPEG still picture compression standard, *IEEE Transactions on Consumer Electronics*, Feb. 1992, vol. 38, no. 1, pp. 18—34.

8. **Sikora T.** MPEG digital video-coding standards, *IEEE Signal Processing Magazine*, Sep. 1997, vol. 14, no. 5, pp. 82—100.

9. **Mulla A., Baviskar J., Baviskar A., Warty C.** Image compression scheme based on zig-zag 3D-DCT and LDPC coding, *Proc. International Conference on Advances in Computing, Communications and Informatics*, 2014, pp. 2380—2384.

10. **Fahmi Sh. S., Almagro M. M., Sokolov Yu. M., Bobrowski A. I., Eid M. M., Salem A.** Precision, speed and complexity of the device coding of images using control points, *Scientific and technical journal of information technologies, mechanics and optics*, 2016, vol. 16, no. 4, pp. 678—689 (in Russian).

11. **Fahmi Sh. S., Almagro M., Salem, A., Bobrovsky A. I., Uom M. M.** Information measure of the quality of encoders at the pivot points, *Questions of radio electronics, ser. Television engineering*, 2016, vol. 3, pp. 86—91 (in Russian).

12. **Fahmi Sh. S., Sokolov Y. M.** Video systems on chip: new solutions in the tasks in video processing, *The XXIV int. science. method. Conf. "Modern education: content, technology, quality Modern education: content, technology, quality*, SPb., SPbGETU "LETI", April 21, 2018, vol. 2, pp. 17—19 (in Russian).

13. **Fahmi Sh. S.** the Concept of designing intelligent transport video systems based on the technology "system on a chip", *Journal of the University of water communications*, 2013, iss. II (XVIII), pp. 79—88 (in Russian).

14. **Moiseev N. N.** Mathematical problems of system analysis. Moscow, Science, 1981, 488 p. (in Russian).

15. **Fahmi Sh. S., Sycolin A. K., Kolesnikov E. I., Occur S. V.** The functionality of the interchange complexity and accuracy of coding systems a continuous signal, *Informacionnye Tekhnologii*, 2011, no. 4, pp. 71—77 (in Russian).

16. **Lee M. C., Chan K. W., Adjeroh D. A.** Quantization of 3D-DCT coefficients and Scan Order for Video Compression, *Journal of visual communication and image representation*, 1997, vol. 8, no. 4, pp. 405—422.

17. **Bozinovic N., Konrad J.** Scan order and quantization for 3D-DCT coding, *S&T/SPIE Symposium on Image and Video Communications and Proc.*, Jul. 8—11, 2003.

18. **Mehanna A., Aggoun A., Abdulfatah O., Swash M. R., Tsekleves E.** Adaptive 3D-DCT based compression algorithms for integral images, *Broadband Multimedia Systems and Broadcasting*, 2013, pp. 1—5.

19. **Priyadarshini K. S., Sharvani G. S.** A survey on parallel computing of image compression algorithms, *Proc. International Conference, Computational Systems for Health & Sustainability*, April 2015, pp. 178—183.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИОМЕДИЦИНСКИХ СИСТЕМАХ
INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

УДК 004.89 DOI: 10.17587/it.26.46-55

Я. Е. Львович, д-р техн. наук, проф., e-mail: office@vivt.ru,
Воронежский институт высоких технологий, Воронеж,
И. Л. Каширина, д-р техн. наук, проф., e-mail: kash.irina@mail.ru,
М. В. Демченко, аспирант, e-mail: masha-vrn@yandex.ru,
Воронежский государственный университет, г. Воронеж

Использование методов машинного обучения для исследования маркеров атеросклероза магистральных артерий

На основе методов интеллектуального анализа данных и машинного обучения анализируются маркеры атеросклероза магистральных артерий. Основная цель данного исследования – поиск факторов и их ассоциаций, определяющих высокую вероятность наличия атеросклероза магистральных артерий на основании данных многоканальной объемной сфигмографии, и разработка прикладного программного обеспечения для ранней диагностики этого заболевания.

Ключевые слова: *машинное обучение, нейронные сети, деревья решений, карты Кохонена, атеросклероз артерий конечностей, бинарная классификация, чувствительность, специфичность*

Введение

Заболевания сердечно-сосудистой системы являются наиболее распространенной причиной смертности во всем мире, вызывающей более 30 % от общего числа всех фатальных исходов. Одной из ведущих причин ишемических поражений органов является такое заболевание, как атеросклероз, т. е. патология, приводящая к деформации и сужению просвета сосудов вплоть до обтурации (закупорки сосуда). Во всем мире непрерывно ведутся исследования, направленные на предупреждение и лечение этого заболевания [1].

Согласно новым исследованиям простым и в то же время эффективным критерием атеросклеротического поражения сосудов является нарушение баланса систолического артериального давления (САД) на верхних и нижних конечностях [2]. В то время как позиции лодыжечно-плечевого индекса (ЛПИ), характеризующего отношение САД на руках и ногах и являющегося маркером атеросклеротического поражения конечностей, в современных руководствах достаточно хорошо определены, значение асимметрии АД на руках (Δ САДр) или ногах (Δ САДн) для диагностики и прогноза атеросклероза еще изучается [2—4]. В качестве коэффициента значимой асимметрии артериального давления на верхних или нижних конечностях многими медицинскими экспертами принимается значение 15 мм рт. ст. Однако данная величина не является общепринятым показателем и часто во многих исследованиях при ее определении встречаются расхождения. В частности, авторы исследования [2] рассматривают значимую разницу давлений 10 мм рт. ст. на верхних и 20 мм рт. ст., на нижних конечностях. Опубликованные в 2014 г. данные Фрамингемского эпидемиологического проекта показали, что разница САД на руках, превышающая 10 мм рт. ст., встречалась в 26,2 % случаев от общего числа наблюдений за жителями старше 40 лет и была связана с повышенным риском развития различных сердечно-сосудистых заболеваний [3]. В одном из современных метаанализов было продемонстрировано, что разница САД на руках, превышающая 15 мм рт. ст., является не только предиктором атеросклеротического поражения периферических артерий, но и значимо ассоциируется с увеличением риска сердечно-сосудистой смертности [4].

В ходе данного исследования с помощью методов машинного обучения предпринята попытка оценить эффективность использования многоканальной объемной сфигмографии (МОС) при проведении скрининга взрослого населения. Выбор данной методики объясняется необходимостью дополнить набор достаточно простых диагностических инструментов (анамнез, антропометрия, шкала SCORE и т. д.) тестом, способным обнаруживать признаки бессимптомного атеросклеротического поражения артерий без существенных дополнительных затрат [5—7].

Исследование включает следующие этапы.

1. Анализ взаимосвязи маркеров атеросклероза (полученных на основе результатов МОС) и их предикторов с использованием методов машинного обучения. Поиск зависимостей маркеров атеросклероза от гемодинамических параметров пациента и построение моделей зависимостей маркеров от социально-демографических, антропометрических и клинических факторов.

2. Отыскание значений асимметрии САД на руках и ногах, являющихся наиболее информативными признаками атеросклероза магистральных артерий.

3. Разработка комплекса программ для диагностики атеросклероза с помощью нейросетевых моделей.

4. Разработка веб-сайта и мобильного приложения для пациентов и врачей, позволяющих пройти ряд диагностических тестов по обнаружению атеросклероза артерий конечностей.

Исходные данные

В данном исследовании для выявления нарушений артериального кровообращения используется комбинация нескольких признаков (маркеров), основанных на значениях САД и представленных следующими формулами:

$$ArmsIndex = \begin{cases} 1, \text{ если } |\Delta\text{САДр}| \geqslant \Delta_1; \\ 0, \text{ иначе;} \end{cases} \quad (1)$$

$$LegsIndex = \begin{cases} 1, \text{ если } |\Delta\text{САДн}| \geqslant \Delta_2; \\ 0, \text{ иначе;} \end{cases} \quad (2)$$

$$ЛПИ = \begin{cases} 1, \text{ если ЛПИл} \leqslant \Delta_3 \text{ или ЛПИп} \leqslant \Delta_3; \\ 0, \text{ иначе.} \end{cases} \quad (3)$$

В формулах (1)—(3) присутствуют три параметра Δ_1, Δ_2, Δ_3, характеризующих диагностически значимое расхождение САД на руках или ногах, а также значение ЛПИ на левых или правых конечностях. Значение этих параметров предлагается определять с помощью алгоритмов машинного обучения, на основе которых будет строиться диагностическая модель, а затем будут отыскиваться параметры, обеспечивающие максимальную достоверность построенной модели [8].

В качестве обучающей выборки для задачи машинного обучения на начальном этапе использовалась неперсонифицированная база данных обследования пациентов Богучарского района Воронежской области, проведенного в рамках программы всеобщей диспансеризации. В ходе программы было обследовано 522 жителя, у которых помимо обязательных процедур методом МОС выполнялось синхронное измерение систолического и диастолического артериального давления на верхних и нижних конечностях, вычислялась их разница, а также автоматически рассчитывалось значение ЛПИ и показателей асимметрии на конечностях. Кроме того, фиксировались антропометрические, клинические, гемодинамические и другие показатели пациентов, что в совокупности составило 45 параметров. Перечень из 28 наиболее значимых показателей представлен в табл. 1. Фрагмент исходной вы-

Таблица 1

Входные признаки

Категория признаков	Признаки
Гемодинамические	Систолическое/диастолическое/пульсовое артериальное давление на правой/левой руке/ноге (САДпр, ДАДпр, ПДпр, САДлр, ДАДлр, ПДлр, САДпн, САДлн), частота сердечных сокращений (ЧСС), скорость пульсовой волны (cfPWV, baPWV)
Социально-демографические	Пол, возраст, курильщик
Антропометрические	Рост, вес, индекс массы тела (ИМТ)
Лабораторные	Глюкоза, общий холестерин (ОХ)
Клинические	Артериальная гипертония (АГ), стенокардия, инфаркт миокарда (ИМ), острое нарушение мозгового кровообращения (ОНМК), аортокоронарное шунтирование/чрескожное вмешательство (АКШ/ЧКВ), сахарный диабет (СД), хроническая сердечная недостаточность (ХСН), фибрилляция/трепетание предсердий (ФП/ТП), ожирение

Таблица 2

Фрагмент исходного набора данных

Код	Возраст	Пол	Курит	АГ	ИМ	ОНМК	СД	ХСН	ФП/ТП	Рост	Вес	Глюкоза	ОХ
1	51	М	Нет	Нет	Да	Нет	Нет	Нет	Нет	174	84	5,14	6,80
2	51	Ж	Нет	Нет	Нет	Нет	Нет	Нет	Нет	152	53	4,26	7,68
3	64	М	Нет	Да	Нет	Да	Да	Да	Нет	165	80	8,50	7,00
4	52	Ж	Нет	Да	Нет	Нет	Нет	Нет	Нет	158	62	4,20	4,50
5	66	Ж	Нет	Да	Нет	Нет	Нет	Нет	Нет	160	90	5,01	4,61
6	50	М	Да	Да	Нет	Нет	Нет	Нет	Нет	172	69	5,70	4,30
7	58	М	Нет	Нет	Нет	Нет	Да	Нет	Да	175	138	11,35	6,40

борки представлен в табл. 2. Детали данного обследования отражены в источниках [9, 10]. Исследование проводилось по инициативе областного кардиологического диспансера при ВОКБ № 1 г. Воронежа.

Исследуемый набор данных представляет собой выборку достаточно малого объема, так как содержит 522 примера, что соответствует числу обследованных пациентов. В условиях малого объема дополнительное разбиение выборки на обучающее и тестовое подмножества способно отрицательно сказаться на ее общей репрезентативности и, следовательно, качестве обучения.

Для решения этой проблемы в данном исследовании была применена процедура n-кратной кросс-проверки в сочетании с механизмом бутстрепа при формировании подвыборок [11]. Такая процедура позволяет, с одной стороны, задействовать весь имеющийся набор данных в обучении, а с другой стороны, максимально точно оценить производительность модели на произвольных наборах данных.

Помимо малого объема другим критичным недостатком исходной выборки, типичным для задач медицинской диагностики и способным отрицательно повлиять на чувствительность диагностических моделей, является ее несбалансированность, выражающаяся в том, что число здоровых пациентов в выборке значительно превосходит число пациентов с атеросклерозом. Классическая стратегия минимизации общего числа ошибок классификации в данном случае способна отрицательным образом сказаться на качестве распознавания класса, представленного меньшинством объектов.

Для решения этой проблемы была задействована матрица стоимости ошибок классификации C, отражающая издержки, связанные со всеми возможными исходами, где C_{ij} — затраты на ошибку отнесения элемента класса i к классу j. Большие веса назначаются тем классам, ошибка по которым является более критичной, в данной задаче — классу пациентов с атеросклерозом. Таким образом, в результате анализа исследуемого набора данных были выявлены проблемы, требующие применения особых методик разработки и обучения классифицирующих моделей.

Метрики качества

Значения метрик рассчитывались на основании матрицы ошибок классификации (табл. 3), строкам которой соответствуют ожидаемые результаты классификации, а столбцам — результаты, предсказанные моделью. Главная и побочная диагонали матрицы содержат число верных (True Positives, True Negatives) и ошибочных ответов (False Positives, False Negatives) соответственно.

На основании истинных и ложных ответов модели можно вычислить множество метрик качества классификации, в частности, Accuracy, Precision, Recall и True Negative Rate, представленных следующими формулами (при этом Recall и True Negative Rate в случае задач медицинской диагностики приобретают смысл чувствительности и специфичности, т.е. характеризуют способность модели верно классифи-

Таблица 3

Матрица ошибок классификации

Фактический класс	Предсказанный класс	
	Класс "+" (больные)	Класс "–" (здоровые)
Класс "+" (больные)	TP	FN
Класс "–" (здоровые)	FP	TN

цировать больных и здоровых пациентов соответственно):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}; \quad (4)$$

$$Precision = \frac{TP}{TP + FP}; \quad (5)$$

$$Recall = \frac{TP}{TP + FN}; \quad (6)$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}. \quad (7)$$

В медицинской диагностике оптимален метод исследования, который был бы как высоко специфичен, так и высоко чувствителен. Однако в реальности это труднодостижимо, так как повышение чувствительности алгоритма, как правило, сопровождается снижением его специфичности и наоборот. В основном выбор между чувствительностью и специфичностью зависит от условий конкретной задачи и от специфики заболевания. Например, в случае дорогого и травматичного лечения критичным становится риск "ложных тревог", однако в то же время пропуск опасных заболеваний на ранних стадиях способствует прогрессированию и развитию потенциально угрожающих здоровью состояний.

Чтобы создать оптимальную диагностическую систему, необходимо найти компромисс между показателями чувствительности и специфичности. Для решения этой проблемы хорошо подходит такой инструмент, как характеристические ROC-кривые. Графическое представление построенных в ходе данного исследования ROC-кривых, отражающих качество некоторых из разработанных классификаторов, представлено на рис. 1.

Графики кривых на рис. 1 отражают взаимосвязь между истинноположительными (TPR, чувствительность) и ложноположительными (FPR, 1-специфичность) результатами классификации, а площадь AUC под кривой является интегральным критерием, позволяющим отнести построенную модель к определенному классу качества, что демонстрирует табл. 4.

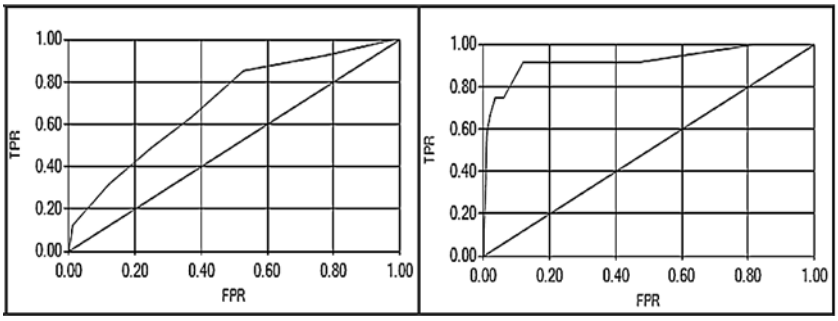


Рис. 1. Характеристические ROC-кривые

Таблица 4

Шкала классификации AUC

Интервал AUC	Качество модели
0,9...1	Отличное
0,8...0,9	Очень хорошее
0,7...0,8	Хорошее
0,6...0,7	Среднее
0,5...0,6	Неудовлетворительное

Площадь под кривой AUC часто рассматривается в задачах медицинской диагностики как оптимальный и достоверный критерий качества.

Метод деревьев решений

С учетом особенностей исходного набора данных в первую очередь в качестве используемого метода машинного обучения были рассмотрены деревья решений, широко применяемые в медицинских исследованиях.

Построение моделей выполнялось с использованием библиотеки машинного обучения Scikit-Learn для языка Python, в которой реализована как настройка весов матрицы стоимости ошибок классификации, так и процедура кросс-проверки.

Среди неоспоримых достоинств метода деревьев решений следует отметить его высокую интерпретируемость, поскольку в качестве результата мы получаем на выходе интуитивно понятную графическую иллюстрацию классификационной модели, которая может быть преобразована в набор правил. Например, одно из правил, построенных для предсказания маркера ЛПИ по социально-демографическим, антропометрическим и клиническим факторам, имеет вид: "Если вес > 114,5 и (возраст > 66,5 или сахарный диабет (СД)), то ЛПИ < 0,9".

Метод деревьев решений позволяет не только разрабатывать классификационные диагностические алгоритмы, но и оценивать значимость используемых признаков.

Важность признака в методе деревьев решений вычисляется как нормализованный итог сокращения критерия ветвления, вызванный этим признаком. Критерий ветвления измеряет меру неопределенности в узлах дерева. В качестве такого критерия обычно используется индекс Джини или информационная энтропия. С помощью данного метода был проведен отбор наиболее значимых признаков, выбранных далее в качестве предикторов, т. е. независимых переменных анализа, на основании которых выполняется предсказание значений маркеров в целях выявления зависимостей между маркерами атеросклероза и социально-демографическими, антропометрическими, клиническими и гемодинамическими параметрами пациента.

На рис. 2 приведены значимые социально-демографические, антропометрические и клинические параметры для маркера ЛПИ (значимость масштабирована в интервал [0;100]).

Благодаря использованному механизму корректировки ошибок для классов, представленных меньшинством объектов, с помощью метода деревьев решений были построены модели, обладающие высокой точностью на обучающей выборке. Однако результаты кросс-проверки выявили сильное ухудшение чувствительности классификационных деревьев в процессе тестирования. В табл. 5 приведены значения чувствительности, специфичности и доли правильных ответов деревьев решений для всех маркеров и набора гемодинамических параметров.

В целях устранения недостатка низкой чувствительности классификационных деревьев было принято решение о необходимости построения модели, обладающей лучшими показателями критериев качества, а именно нейронной сети архитектуры многослойный персептрон, обученной по алгоритму обратного распространения ошибки [13]. При этом метод деревьев решений позволил существенно снизить размерность признакового пространства для построения нейросетевых моделей.

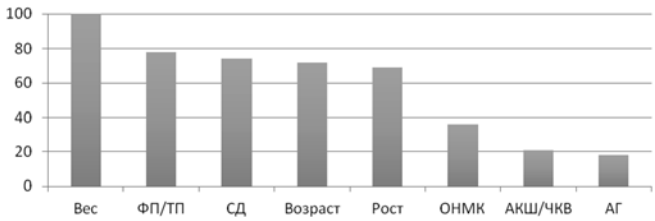


Рис. 2. Значимость предикторов для ЛПИ

Таблица 5

Качество деревьев решений по результатам кросс-проверки

Маркер	Чувствительность, % Recall	Специфичность, % True Negative Rate	Доля верных ответов, % Accuracy
ArmsIndex	39,7	94,7	90
LegsIndex	25,2	92,8	87,9
ЛПИ	35	97,6	92,5

Нейронные сети архитектуры MLP

Нейронная сеть архитектуры многослойный персептрон (MLP) состоит из нескольких слоев нейронов, причем каждый нейрон предыдущего слоя связан с каждым нейроном последующего слоя. Во многих практических приложениях оказывается достаточным рассмотрение двухслойной нейронной сети, имеющей скрытый и выходной слои. Именно такая конфигурация сети была использована для моделирования зависимостей маркеров атеросклероза от признаков, отобранных на этапе построения деревьев решений.

Классический алгоритм обучения сетей архитектуры MLP ставит задачу минимизации целевой функции ошибок сети. В задачах классификации, как правило, в качестве функции потерь используется кросс-энтропия (перекрестная энтропия), которая характеризует отклонение правильного ответа $y^*(x)$ от полученного сетью ответа y.

К сожалению, стандартный алгоритм обратного распространения ошибки не позволяет достигнуть желаемой чувствительности сети на несбалансированных выборках. Для решения данной проблемы во многих исследованиях предлагаются различные модификации классического алгоритма [12], одна из которых была использована в данной работе.

Модификация алгоритма заключалась во внедрении дополнительных конфигурационных параметров — коэффициентов матрицы затрат, настраиваемых пользователем наряду с такими основными параметрами, как скорость обучения, момент, число эпох. На основании матрицы затрат формировались коэффициенты целевой функции, являющиеся корректирующими факторами. Таким образом, стандартная функция ошибок приобретала смысл функции затрат, гибко учитывающей даже ошибки малочисленных классов. Основные модификации отражены в следующих формулах:

$$CostFunction = -\sum_{j=1}^{m} y_{kj}^{*} \log y_{kj} K[class(k), j]; \quad (8)$$

$$K[i, j] = \begin{cases} CostVector[i], & i = j; \\ Cost[i, j], & i \neq j; \end{cases} \quad (9)$$

$$Cost[i, j] = \begin{cases} c_{ij}, & i \neq j; \\ 0, & i = j; \end{cases} \quad (10)$$

$$CostVector[i] = \begin{cases} 0, & \text{если} \quad P(i) = 1; \\ \dfrac{1}{1 - P(i)} \sum\limits_{j \neq i} P(j) Cost[i, j], & \text{иначе}, \end{cases} \quad (11)$$

где *CostFunktion* — функция затрат; m — число выходов сети; y_{kj}^{*} — правильное значение j-го выхода сети на k-м входном векторе; y_{kj} — полученное значение j-го выхода сети на k-м входном векторе; $K[i, j]$ — корректирующий фактор; $Cost[i, j]$ — матрица затрат на ошибку определения класса i к классу j; $CostVector[i]$ — вектор затрат ошибочного определения объекта класса i к любому классу, отличному от i; $P(i)$ — априорная вероятность принадлежности объекта классу i; $class(k)$ — ожидаемый класс k-го примера выборки.

В качестве инструмента программной реализации была выбрана библиотека, реализованная на платформе.Net — Aforge.Neuro, содержащая набор базовых функций, необходимых для работы с различными нейросетевыми архитектурами. На базе платформы .Net было разработано приложение для диагностики атеросклероза с помощью нейронных сетей архитектуры MLP, пример работы которого отражен на рис. 3.

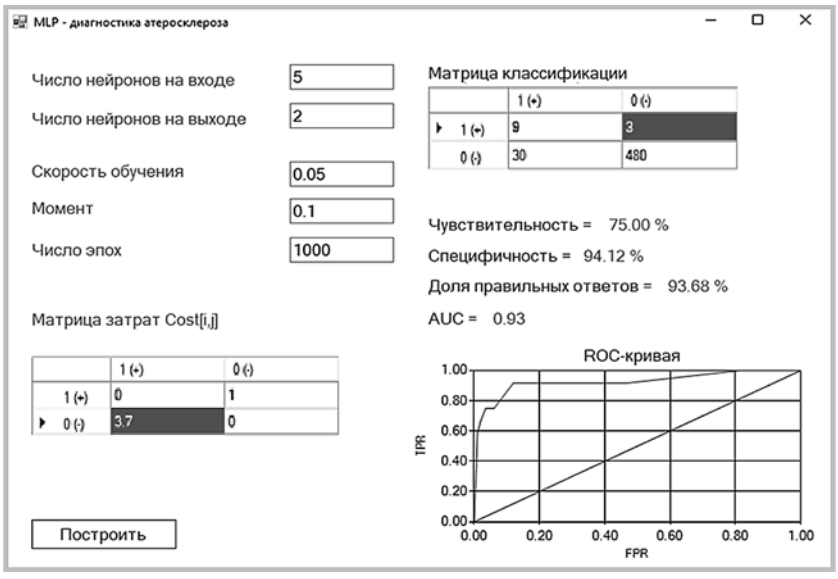


Рис. 3. Диагностика атеросклероза с помощью нейронных сетей MLP

Помимо основного обучающего модуля разработанный программный продукт предоставляет функционал определения основных критериев качества нейросетевых моделей, а также построения ROC-кривых.

На рис. 3 представлены результаты построения нейронной сети для маркера ЛПИ по набору гемодинамических параметров. Проиллюстрированная модель обладает высокой чувствительностью (75 %), специфичностью (94,12 %) и долей верных ответов (93,68 %), что в целом обеспечивает высокое качество по шкале AUC — 0,93.

Таким образом, с помощью применения классических подходов нейросетевого обучения и с учетом оптимизации базового алгоритма удалось достигнуть высокой эффективности построенных классификаторов. Однако очевидным недостатком построенных нейросетевых моделей является их низкая интерпретируемость.

Карты Кохонена

Методом, в определенном смысле объединяющим преимущества нейросетевых и интерпретируемых алгоритмов и предоставляющим возможность визуализации закономерностей между маркерами и предикторами, обнаруженных в исходном наборе данных, является метод самоорганизующихся карт Кохонена [11]. Существенным отличием карт Кохонена от рассмотренных ранее сетей MLP и деревьев решений является то, что данный метод представляет собой метод нейросетевой кластеризации (а не классификации). Данная сеть обучается без учителя на основе алгоритма самоорганизации (Self Organizing Map).

Для построения карт Кохонена для каждого из маркеров по набору гемодинамических, клинических и других показателей также был разработан программный продукт, выполненный на технологиях .NET [13]. Набор раскрасок карт Кохонена по входным признакам позволяет выявить ряд закономерностей в исходном наборе данных.

Например, при построении карты Кохонена для исследования связи

Таблица 6

Характеристики нейронных сетей для маркера ArmsIndex

№ сети	Δ_1	Набор факторов	Чувствительность	Специфичность	Доля верных ответов	AUC
1	14	Антропометрические и социально-демографические	*75,61*	*74,45*	*85,33*	*0,87*
2	14	Клинические	73,41	72,14	81,46	**0,84**
3	13	Гемодинамические	70,73	65,93	78,09	**0,74**

маркера ЛПИ с отобранными ранее социально-демографическими, антропометрическими и клиническими факторами сформировалось семь кластеров, в одном из которых наблюдалось значимое повышение числа пациентов с низким значением ЛПИ. В данном кластере наблюдались повышенные значения показателей веса (среднее значение — 92 кг), возраста (в среднем — 60 лет), наличие в анамнезе стенокардии (80 % пациентов), сахарного диабета (85 % пациентов), хронической сердечной недостаточности (74 % пациентов). При построении карт Кохонена по признакам, значимым для маркера LegsIndex, также было выделено семь кластеров, в одном из которых наблюдалось значимое повышение LegsIndex. В данном кластере преобладали данные лиц мужского пола (90 % пациентов), со статусом курильщика (65 % пациентов), с отягощенным анамнезом (67 % пациентов), артериальной гипертонией (87 % пациентов), фибрилляцией/трепетанием предсердий (56 % пациентов), хронической сердечной недостаточностью (56 % пациентов). При этом разброс значений внутри кластеров (коэффициент вариации) не превышал 20 %, что позволяет считать кластеры достаточно однородными. Таким образом, карты Кохонена позволяют не только предсказывать риск атеросклероза для новых пациентов, но и позволяют сформировать обобщенный портрет пациента с наличием того или иного маркера атеросклероза. Подробно результаты этого анализа описаны в работе [13], там же представлены результаты визуализации.

Определение коэффициентов асимметрии артериального давления

На начальном этапе анализа в качестве коэффициентов асимметрии артериального давления были выбраны значения $\Delta_1 = \Delta_2 = 15$ мм рт. ст., $\Delta_3 = 0{,}9$, предложенные экспертами области медицины и представленные в формулах (1) и (2). Однако если значение Δ_3 является хорошо установленным в медицинской практике, то Δ_1 и Δ_2 не являются общепринятыми показателями и нередко во многих исследованиях при их определении встречаются расхождения. В рамках данного исследования была проведена калибровка данных величин.

Решение этой задачи подразумевало построение набора нейронных сетей различных конфигураций с зависимыми переменными-маркерами, сформированными с учетом различных значений асимметрии от 10 до 15. В табл. 6 отражен набор нейронных сетей наиболее высокого качества для маркера ArmsIndex, построенных по различным наборам предикторов (гемодинамических, клинических, антропометрических и др.).

Все модели, характеристики которых приведены в табл. 5, отличаются по значениям основных метрик качества — чувствительности, специфичности, доли правильных ответов и AUC, однако по причине того, что в качестве наиболее приоритетного критерия медицинскими экспертами был выбран интегральный критерий AUC, наиболее достоверное значение коэффициента асимметрии соответствовало сети с максимальным значением этого критерия.

Для маркера ArmsIndex наиболее информативным коэффициентом асимметрии было выбрано значение 14 (у сети с максимальным значением AUC — 0,87), для LegsIndex — 15. В табл. 7 приведены характеристики наиболее качественных моделей, построенных для каждого из рассматриваемых маркеров.

Максимального качества по кросс-проверке (AUC = 0,93) удалось достигнуть нейронной сети для маркера ЛПИ, построенной по набору гемодинамических параметров. В набор гемодинамических признаков входят признаки давления, ЧСС и скорость каротидно-феморальной пульсовой волны. Поскольку повышение скорости пульсовой волны является общепринятым фактором развития атеросклероза, его

Таблица 7

Нейронные сети наилучшего качества по метрике AUC ROC

Маркер	Набор предикторов	Качество по кросс-проверке
ЛПИ	САДпр, САДлн, ЧСС, ДАДпр, cfPWV_calc> 10 м/с	0,93 (отличное)
ArmsIndex (Δ_1 = 14)	Пол, Анамнез, АГ, СД, ХСН, Возраст, Рост, Вес	0,8 (очень хорошее)
LegsIndex (Δ_2 = 15)	ЧСС, САДлн, САДпр, САДлр, cfPWV_calc > 10 м/с, ДАДлр, ДАДпр	0,73 (хорошее)
ЛПИ	Возраст, Рост, Вес, АГ, ОНМК, АКШ/ЧКВ, СД, ФП/ТП	0,77 (хорошее)

тесная связь с ЛПИ подтверждает эффективность ЛПИ в качестве маркера атеросклероза конечностей. Также необходимо отметить, что нейронные сети для ЛПИ обладали высоким качеством на наборе социально-демографических, антропометрических и клинических предикторов. Модель по этим параметрам проявила высокую чувствительность (более 81 %) и в целом является моделью хорошего качества по критерию AUC (0,77). Ценность такой модели заключается в том, что ее возможно применять в качестве инструмента быстрой диагностики, без проведения дополнительных лабораторных исследований, измерений артериального давления и инвазивного вмешательства, что позволяет использовать ее в системах первичной дистанционной диагностики. Также модель хорошего качества удалось построить для ArmsIndex (Δ_1 = 14) и группы предикторов, включающих пол, возраст, рост, вес, отягощенный анамнез, заболевания АГ, ХСН и СД.

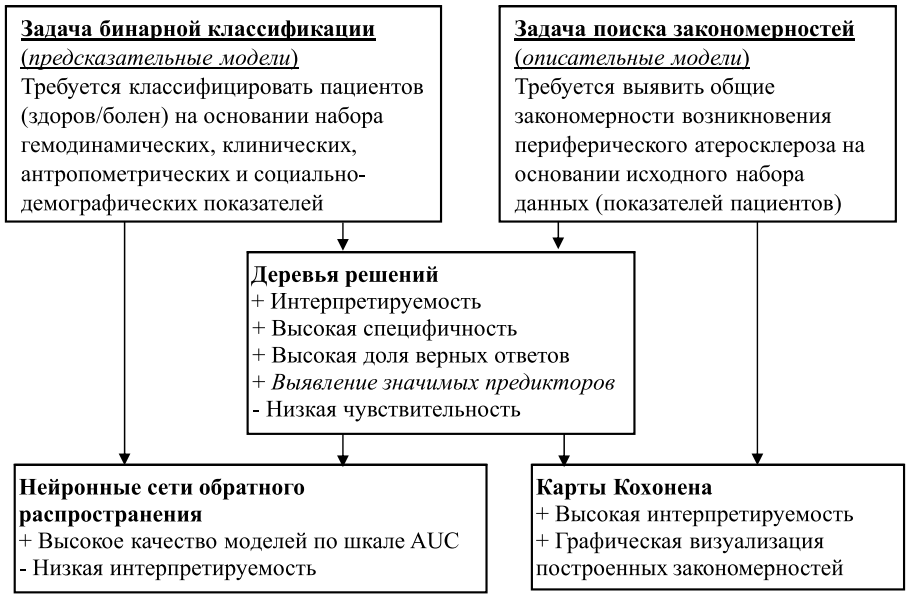


Рис. 4. Взаимосвязь использованных методов анализа

Результаты исследования

По результатам данного исследования можно сделать следующие выводы:

1. Подтверждена эффективность ЛПИ как важного маркера атеросклероза магистральных артерий.

2. Разработана нейросетевая модель диагностики атеросклероза по антропометрическим, клиническим и социально-демографическим признакам пациента, позволяющая выявлять риск заболевания с точностью более 77 %, не требуя лабораторной диагностики и инвазивных вмешательств для использования в дистанционных диагностических системах.

3. С помощью карт Кохонена установлены общие закономерности риска атеросклероза, подтверждающие и дополняющие общепринятые признаки развития данного заболевания.

4. Проведен всесторонний анализ взаимосвязи маркеров атеросклероза магистральных артерий, полученных на основе результатов МОС и их предикторов. Задачи, исследованные в процессе анализа, и методы их решения представлены на рис. 4.

5. Вычислены наиболее информативные коэффициенты асимметрии артериального давления на руках. С учетом чувствительности и специфичности построенных моделей можно рекомендовать последовательность использования маркеров атеросклероза (рис. 5).

6. Наряду с представленными программными модулями для выявления маркеров атеросклероза с помощью методов машинного обучения было также разработано веб-приложение для пациентов и врачей, предназначенное для онлайн-диагностики атеросклероза, которое размещено по адресу: http://cardio-002-site1.dtempurl.com/. Данный ресурс, во-первых, содержит основную информацию о проведенном анализе и его результатах, а также ряд диагностических тестов для определе-

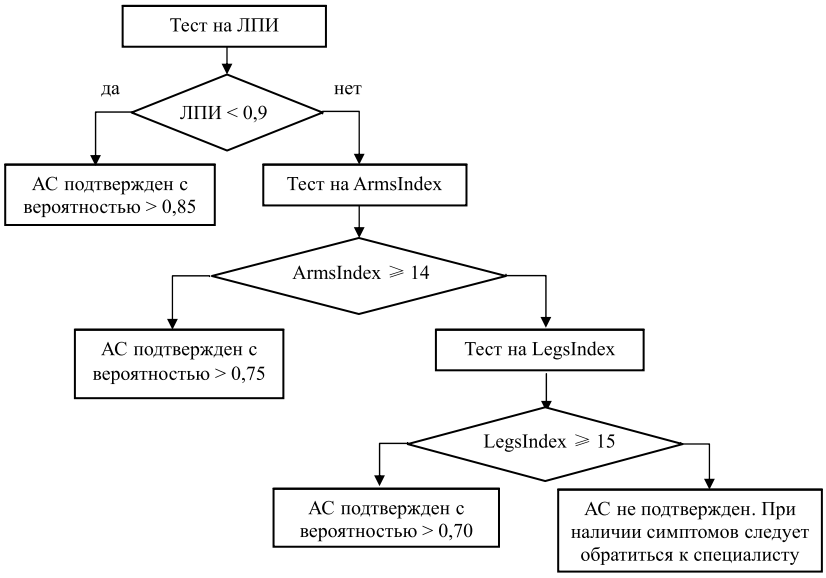


Рис. 5. Схема диагностики атеросклероза магистральных артерий с учетом полученной калибровки пороговых значений маркеров

ния риска атеросклероза, сформированных на основе выявленных закономерностей.

Заключение

Таким образом, в результате данного исследования был проведен всесторонний анализ маркеров атеросклероза и их предикторов, определены значения асимметрии систолического АД на руках и ногах, являющиеся наиболее информативными признаками атеросклероза магистральных артерий, а также разработан комплекс программ для диагностики атеросклероза с помощью нейросетевых моделей, в том числе веб-сайт для пациентов и врачей, позволяющий пройти ряд диагностических тестов в целях дистанционной оценки риска атеросклероза артерий конечностей.

Список литературы

1. **Tuzcu E. M., Kapadia S. R., Tutar E.** et al. High prevalence of coronary atherosclerosis in asymptomatic teenagers and young adults: evidence from intravascular ultrasound // *Circulation*. 2001;103:2705—2710.

2. **Mitchell E.** Noninvasive diagnosis of arterial disease. UpToDate. URL: https://www.uptodate.com/contents/noninvasive-diagnosis-of-arterial-disease.

3. **Weinberg I.** et al. The Systolic Blood Pressure Difference Between Arms and Cardiovascular Disease in the Framingham Heart Study // The American Journal of Medicine. 2014:127; 209—215.

4. **Clark C. E., Taylor R. S., Shore A. C.** et al. Association of a difference in systolic blood pressure between arms with vascular disease and mortality: a systematic review and meta-analysis // Lancet. 2012:380 (9838):218.

5. **Sheng C. S., Liu M., Zeng W. F.** et al. Four-Limb Blood Pressure as Predictors of Mortality in Elderly Chinese // Hypertension. 2013: 61: 1155—60.

6. **Su H.-M., Lin T.-H., Hsu P.-C.** et al. Association of Interarm Systolic Blood Pressure Difference with Atherosclerosis and LeftVentricular Hypertrophy // PLoS ONE. 2012; 7(8): e41173.

7. **Su H.-M., Lin T.-H., Hsu P.-C.** et al. Association of Bilateral Brachial-Ankle Pulse Wave Velocity Difference with Peripheral VascularDisease and Left Ventricular Mass Index // PLoS ONE. 2014;9(2): e88331.

8. **Демченко М. В., Каширина И. Л.** Сравнительный анализ и оценка эффективности маркеров атеросклероза магистральных артерий // Сб. тр. Междунар. науч.-техн. конф. "Актуальные проблемы прикладной математики, информатики и механики" (Воронеж, 18—20 декабря 2017 г.). Воронеж, 2017. С. 636—644.

9. **Хохлов Р. А., Остроушко Н. И., Гайдашев А. Э., Кирсанов Д. В., Ахмеджанов Н. М.** Предикторы атеросклеротического поражения артерий конечностей по данным кардиоангиологического скрининга взрослого населения // Рациональная Фармакотерапия в Кардиологии. 2015. № 11(5). С. 470—476.

10. **Хохлов Р. А., Гайдашев А. Э., Ахмеджанов Н. М.** Использование многоканальной объемной сфигмографии для кардиоангиологического скрининга взрослого населения // Рациональная фармакотерапия в кардиологии. 2015. № 11(4). С. 470—476.

11. **Каширина И. Л., Азарнова Т. В.** Нейросетевые и гибридные системы: учебно-методическое пособие для вузов. Воронеж: Издательский дом ВГУ, 2014. 80 с.

12. **Kukar M., Kononenko I.** Cost-Sensitive Learning with Neural Networks // Machine Learning and Data Mining: proceedings of the 13th European Conference on Artificial Intelligence. 1998. P. 445—449.

13. **Демченко М. В., Каширина И. Л.** Применение самоорганизующихся карт Кохонена в задаче диагностики атеросклероза // Матер. XVIII Междунар. науч.-метод. конф. "Информатика: проблемы, методология, технологии" (Воронеж, 8—9 февраля 2018 г.). Воронеж, 2018. С. 141—147.

14. **2013 ESH/ESC** Guidelines forthe management of arterial hypertension // Journal of Hypertension. 2013; 31:1281—357.

15. **European** Guidelines on cardiovascular disease prevention in clinical practice (version 2012) // **EurHeart J.** 2012; 33:1635—701.

16. **Criqui M. H., McClelland R. L., McDermott M. M.** et al. The Ankle-Brachial Index and Incident Cardiovascular Events in the MESA (Multi-Ethnic Study of Atherosclerosis) // J Am Coll Cardiol. 2010;56: 1506—12.

17. **ESC** Guidelines on the diagnosis and treatment of peripheral artery diseases: Document covering atherosclerotic disease of extracranial carotid and vertebral, mesenteric, renal, upper and lower extremity arteries: the Task Force on the Diagnosis and Treatment of Peripheral Artery Diseases of the European Society of Cardiology (ESC) // European Heart Journal. 2011; 32, 2851—2906.

18. **Jamei M., Nisnevich A., Wetchler E., Sudat S., Liu E., Upadhyaya K.** Predicting all-cause risk of 30-day hospital readmission using artificial neural networks // PLoS ONE. 2017. 12(7): e0181173. URL: https://doi.org/10.1371/journal.pone.0181173.

Y. E. Lvovich, DSc., Professor, e-mail: office@vivt.ru,
Voronezh Institute of High Technologies, Russian Federation, 394043, Voronezh,
I. L. Kashirina, DSc., Professor, e-mail: kash.irina@mail.ru,
M. V. Demchenko, Post-Graduate Student, e-mail: masha-vrn@yandex.ru,
Voronezh State University Russian Federation, 394036, Voronezh

The Use of Machine Learning Methods to Study Markers of Atherosclerosis of the Great Arteries

This study is devoted to the deep investigation of the main atherosclerosis markers and their predictors using the different machine learning methods. The main goal of the investigation is to find the most efficient markers and predictors, which would allow to identify the presence of this disease with the high accuracy. The dataset under consideration is based on the real patients characteristics, including blood pressure values, clinical, anthropometrical and social features. Such methods as classification trees, multilayered neural networks and Kohonen maps are used for solving the considered problem of binary classification. Accuracy, precision, recall (sensitivity) and true negative rate (specifity) are the main metrics, which are used to evaluate the quality of the resulting classification models. Classification trees method allowed not only to generate the most interpretable models, but also to order the main predictors by their importance, and the most important predictors were then passed as inputs to the multilayered neural networks. The base backpropagation algorithm, used by default for learning the network, has been modified to use the misclassification costs, which allowed to increase the quality of the classification for the unbalanced dataset. In order to show the common rules found in the dataset, the Kohonen maps were build for each of the main atherosclerosis markers. All the developed classification models can be considered as instruments of the non-invasive atherosclerosis diagnostics.

Kyewords: *machine learning, neural networks, multilayered perceptron, classification trees, Kohonen maps, atherosclerosis, binary classification, medical diagnostics, sensitivity, specifity*

DOI: 10.17587/it.26.46-55

References

1. **Tuzcu E. M., Kapadia S. R., Tutar E.** et al. High prevalence of coronary atherosclerosis in asymptomatic teenagers and young adults: evidence from intravascular ultrasound, *Circulation,* 2001, vol. 103, pp. 2705—2710.

2. **Mitchell E.** Noninvasive diagnosis of arterial disease. UpToDate. URL: https://www.uptodate.com/contents/noninvasive-diagnosis-of-arterial-disease.

3. **Weinberg I.** et al. The Systolic Blood Pressure Difference Between Arms and Cardiovascular Disease in the Framingham Heart Study, *The American Journal of Medicine*, 2014, vol. 127, pp. 209—15.

4. **Clark C. E., Taylor R. S., Shore A. C.** et al. Association of a difference in systolic blood pressure between arms with vascular disease and mortality: a systematic review and meta-analysis, *Lancet*, 2012, vol. 379, pp. 905—14.

5. **Sheng C. S., Liu M., Zeng W. F.** et al. Four-Limb Blood Pressure as Predictors of Mortality in Elderly Chinese, *Hypertension*, 2013, vol. 61, pp. 1155—60.

6. **Su H.-M., Lin T.-H., Hsu P.-C.** et al. Association of Interarm Systolic Blood Pressure Difference with Atherosclerosis and LeftVentricular Hypertrophy, *PLoS ONE,* 2012, vol. 7, pp. e41173.

7. **Su H.-M., Lin T.-H., Hsu P.-C.** et al. Association of Bilateral Brachial-Ankle Pulse Wave Velocity Difference with Peripheral VascularDisease and Left Ventricular Mass Index, *PLoS ONE*, 2014, vol. 9, pp. e88331.

8. **Demchenko M. V., Kashirina I. L.** The comparative analysis and estimation of the efficiency of the main atherosclerosis markers, *Aktual'nye Problemy Prikladnoj Matematiki, Informatiki i Mehaniki*, 2017, pp. 636—644 (in Russian).

9. **Hohlov R. A., Ostroushko N. I., Gajdashev A. Je., Kirsanov D. V., Ahmedzhanov N. M.** The predictors of the atherosclerosis of the lower extremities based on the data of the adults cardiological screening, *Racional'naja Farmakoterapija v Kardiologii,* 2015, no. 11(5), pp. 470—476 (in Russian).

10. **Hohlov R. A., Gajdashev A. Je., Ahmedzhanov N. M.** The using of the multichannel three-dimensional sphygmography in the adults cardiological screening, *Racional'naja Farmakoterapija v Kardiologii*, 2015, no. 11(4), pp. 470—476 (in Russian).

11. **Kashirina I. L., Azarnova T. V.** Nejrosetevye i gibridnye sistemy: uchebno-metodicheskoe posobie dlja vuzov (Neural networks and hybrid systems: the manual for the universities), Voronezh, Publishing house of *VGU*, 2014, 80 p. (in Russian).

12. **Kukar M., Kononenko I.** Cost-Sensitive Learning with Neural Networks, *Machine Learning and Data Mining: proceedings of the 13th European Conference on Artificial Intelligence*, 1998, pp. 445—449.

13. **Demchenko M. V., Kashirina I. L.** Using Kohonen maps in the diagnostics of the atherosclerosis, *Materialy XVIII Mezhdunarodnoj nauchno-metodicheskoj konferencii*, 2018, pp. 141—147 (in Russian).

14. **2013 ESH/ESC** Guidelines for the management of arterial hypertension, *Journal of Hypertension,* 2013, vol. 31, pp. 1281—357.

15. **European** Guidelines on cardiovascular disease prevention in clinical practice (version 2012), *European Heart Journal*, 2012, vol. 33, pp. 1635—701.

16. **Criqui M. H., McClelland R. L., McDermott M. M.** et al. The Ankle-Brachial Index and Incident Cardiovascular Events in the MESA (Multi-Ethnic Study of Atherosclerosis), *J Am Coll Cardiol*, 2010, vol. 56, pp. 1506—12.

17. **ESC** Guidelines on the diagnosis and treatment of peripheral artery diseases: Document covering atherosclerotic disease of extracranial carotid and vertebral, mesenteric, renal, upper and lower extremity arteries: the Task Force on the Diagnosis and Treatment of Peripheral Artery Diseases of the European Society of Cardiology (ESC), *European Heart Journal*, 2011, vol. 32, pp. 2851—2906.

УДК 519.237.8 + 576.08 DOI: 10.17587/it.26.56-61

А. И. Лепский, alexlep97@gmail.com,
Санкт-Петербургский государственный университет

Сравнительный анализ алгоритмов кластеризации лейкоцитов по FS и SS параметрам при цитофлуориметрическом исследовании крови

При проведении лабораторных исследований в биологии и медицине большое практическое значение имеет получение формальных правил для оценки численных значений эмпирических данных. Одной из нерешенных задач при цитомертическом исследовании крови является автоматическая типологизация лейкоцитов по размеру и сложности внутриклеточной структуры. Возможным подходом в этом случае может быть применение методов кластерного анализа. Однако при кластеризации белых клеток крови по указанным выше параметрам остается много нерешенных вопросов. В статье исследованы различные алгоритмы кластеризации. При проведении численных экспериментов было показано, что иерархические методы и метод K-средних не дают положительных результатов. Для дальнейшего изучения вопросов, связанных с автоматической типологизацией лейкоцитов крови, наиболее перспективным является метод DBSCAN. Для проведения численных экспериментов был создан программный код, написанный на языке Python.

Ключевые слова: *кластерный анализ, проточная цитометрия, машинное обучение*

За иммунитет организма, как способ его защиты от различных патогенных воздействий, отвечают лейкоциты (белые клетки крови). Их можно разделить на три основные группы: гранулоциты (содержащие крупные сегментированные ядра и имеющие специфическую зернистость цитоплазмы), лимфоциты и моноциты. Две последние группы клеток имеют простое несегментированное ядро и небольшую зернистость цитоплазмы [1].

Математическая иммунология как научная дисциплина начала формироваться во второй половине XX века. В 1964 г. Ф. Барнет и Н. Йерне сформулировали клонально-селекционную теорию иммунитета [2]. В начале 70-х гг. прошлого века были предложены первые математические модели иммунного ответа, построенные согласно этой теории. Они представляли собой системы четырех обыкновенных дифференциальных уравнений и описывали эволюцию В-лимфоцитов в ходе иммунного ответа [3,4]. Большую роль в развитии отечественной математической иммунологии сыграли работы академика Г. И. Марчука. Его модели включали до 15 обыкновенных дифференциальных уравнений [5, 6].

Из последних работ в данном направлении можно отметить систему, построенную С. Р. Кузнецовым, которая содержит не только обыкновенные дифференциальные уравнения, но и уравнения в частных производных первого порядка [7]. Модификации этой модели позволяют изучать динамику пролиферации и дифференцировки неоднородной клеточной популяции [8, 9], а также процессы формирования иммунной памяти Т-лимфоцитов [10].

Экспериментальной основой для построения математических моделей иммунного ответа являются результаты исследования крови. Самым эффективным подходом для получения таких данных сейчас является проточная цитофлуориметрия [11].

Проточный цитофлуориметр — прибор для измерения оптических свойств клеток. Детектор прямого светорассеяния (*forward scatter, FS)* определяет их размер. Детектор бокового светорассеяния (*side scatter, SS*) позволяет судить о сложности внутреннего строения клет-

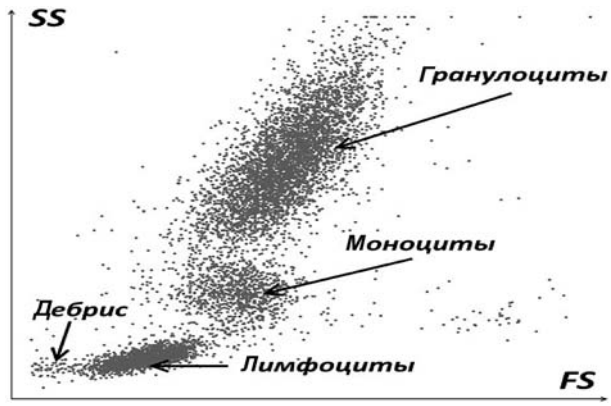


Рис. 1. Распределение лейкоцитов по размеру и сложности строения клеток

ки (соотношение между ядром и цитоплазмой, наличие гранул и других внутриклеточных включений). Используя *FS* и *SS* параметры, можно провести первичный анализ популяций лейкоцитов по размеру и сложности внутреннего строения клеток.

На рис. 1 изображена характерная картина распределения белых клеток крови в осях *FS* и *SS*. Лимфоциты являются самыми маленькими клетками с простым внутренним строением, моноциты крупнее и сложнее по внутреннему строению, гранулоциты являются самыми сложными и самыми большими лейкоцитами. Важно отметить еще одну группу точек, расположенную на рис. 1 ниже и левее всех остальных клеток, это так называемый дебрис (осколки клеток).

Кластерный анализ субпопуляций лейкоцитов

Нерешенной проблемой остается задача автоматической типологизации белых клеток крови, так как гейтирование различных групп (субпопуляций) лейкоцитов при цитометрическом исследовании проводится вручную [11, 12]. Согласно общепринятому соглашению: "Гейт (от англ. *gate* — ворота) — это инструмент для выделения из всего массива полученных данных отдельных популяций клеток, удовлетворяющих определенным условиям" [12, с. 31]. В настоящие время для создания гейтов используются программные средства, позволяющие оператору графически выделять группы клеток на экране консоли (выделение прямоугольного региона, выделение полигонального региона, выделение эллиптического региона) [12, с. 32], что в свою очередь обусловливает субъективность эксперимента и большие значения погрешностей при вычислении числа клеток в различных субпопуляциях.

В связи с этим возникает еще одна проблема. Доказательная медицина — это процесс систематического пересмотра, оценки и использования результатов клинических исследований в целях оказания оптимальной медицинской помощи пациентам, который сочетает в себе определенные принципы и методы. Благодаря их действию инструкции и стратегии в медицине основываются на текущих подтверждающих данных об эффективности разных форм лечения и медицинских услуг в целом. В числе прочего эти принципы предъявляют повышенные требования к качеству лабораторных исследований [13].

Неоднократно в различных медицинских учреждениях и у нас в стране и за рубежом проводился один и тот же эксперимент: нескольким исследователям предлагалось выполнить гейтирование клеток из одной и той же пробы, и всегда получались разные результаты, максимальная погрешность достигала 30 % [14—17].

Свой вклад в погрешность результатов обработки анализов крови вносят два вида ошибок: случайные и системные.

Случайные ошибки сопутствуют любому измерению, как бы тщательно оно не проводилось, и проявляются в некотором различии результатов измерения одного и того же элемента, выполненного данным методом. Они обусловлены в том числе точностью работы персонала лаборатории (неточное считывание результатов, ошибка утомления, неверный подбор класса точности инструментов, психологическая ошибка, например, оказание предпочтения каким-либо цифрам и т. д.)

Системные ошибки зависят от применяемых приборов и реактивов, определяются точностью приборов, происходят от неправильного или неточного выполнения операции, зависят от личных способностей оператора, его органов чувств, привычек и т. д.

Субъективный фактор имеет большое значение и при случайных, и при системных ошибках, поэтому для улучшения качества некоторых видов медицинских услуг необходима разработка машинных (формальных) методов гейтирования клеток крови. Возможным подходом к решению этой задачи может быть кластерный анализ результатов цитометрического исследования.

Произвольный алгоритм кластеризации является отображением

$$A : \begin{cases} X \to \mathbb{N}, \\ \bar{x}_i \mapsto k, \end{cases}$$

которое ставит в соответствие любому элементу $\bar{x}_i$ из некоторого множества X единственное натуральное число k, являющееся номером кластера, которому принадлежит $\bar{x}_i$. Процесс кластеризации разбивает X на попарно дизъюнктные подмножества X_h, которые называются кластерами:

$$X = \bigcup_{h=1}^{m} X_h,$$

где для $\forall h,\ l \mid 1 \leqslant h,\ l \leqslant m$: $X_h \cap X_l = \varnothing$.

Отображение A задает на X отношение эквивалентности. В качестве независимых представителей классов эквивалентности выбирают элементы, называемые центроидами. В n-мерном евклидовом пространстве E^n координаты центроидов равны среднему арифметическому соответствующих координат всех элементов (векторов), входящих в кластер (класс эквивалентности). Если отождествить каждый вектор из E^n с материальной точкой единичной массы, то центроиды можно рассматривать как центры масс кластеров [18].

Число кластеров в некоторых случаях известно заранее, однако чаще всего ставится задача определить оптимальное число кластеров с точки зрения того или иного критерия качества кластеризации.

Для оценки качества кластеризации будем использовать *Silhouette Coefficient* [19, 20]. Коэффициент силуэта не предполагает знания истинных меток объектов и позволяет оценить качество кластеризации, используя только саму (неразмеченную) выборку и результат кластеризации. Сначала силуэт определяется отдельно для каждого объекта. Обозначим a — среднее расстояние от данного объекта до объектов из того же кластера, b — среднее расстояние от данного объекта до объектов из ближайшего кластера (отличного от того, в котором лежит сам объект). Тогда силуэтом данного объекта называется величина

$$s = \frac{b - a}{\max(a, b)}.$$

Силуэтом выборки называется среднее значение силуэта объектов данной выборки.

С помощью силуэта можно выбирать оптимальное число кластеров (если оно заранее неизвестно) — выбирается число кластеров, максимизирующее значение силуэта. Силуэт зависит от формы кластеров и достигает больших значений на более выпуклых кластерах, получаемых с помощью алгоритмов, основанных на восстановлении плотности распределения.

В статье сравниваются результаты численных экспериментов кластеризации точек на евклидовой плоскости с помощью четырех различных методов, с учетом специфики распределения параметров лейкоцитов крови в осях *FS* и *SS*.

Численное моделирование проводили с помощью программного кода, написанного на языке программирования *Python* 3.7.2. В качестве интегрированной среды разработки использовали оболочку *PyCharm*, разработанную для языка программирования *Python* компанией *JetBrains* на основе *IntelliJ IDEA*. *PyCharm* предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов. Для выполнения процедур, реализующих алгоритмы кластеризации, подключались библиотеки *sklearn* и *scipy*. Предварительную стандартизацию данных не проводили, так как оба признака расположены примерно в одном диапазоне. Подбор оптимальных параметров осуществлялся по сетке $k = [1, ..., n]$ (где n — общее число точек) для метода одиночной связи, K-средних и *EM*, и по сетке $n = [1, ..., 10]$; $e = [1, ..., 100]$ (число соседей и радиус) для метода *DBSCAN*. Оптимальным результатом считался тот, который был получен благодаря параметрам, максимизирующим значение коэффициента силуэта.

Метод одиночной связи

Под иерархическими методами кластеризации понимается группа алгоритмов, направленных на построение дерева вложенных кластеров, новые классы эквивалентности создаются путем объединения более мелких кластеров и, таким образом, их дерево формируется от листьев к стволу.

Сначала представим результаты для метода одиночной связи [21—23], где за расстояние между кластерами принимается дистанция между центроидами. Численное моделирование процесса кластеризации лейкоцитов проводили с помощью процедуры *sklearn.cluster.AgglomerativeClustering*. На рис. 2 (см. четвертую сторону обложки) представлена типичная картина формирования кластеров этим методом.

Среднее время работы программы составляет 8,64 с. Этот алгоритм удовлетворительно показал себя на самых разнообразных данных. Из недостатков стоит отметить, что приемлемый результат достигается при достаточно большом числе кластеров. Поэтому помимо трех классов, выделяющих лимфоциты, моноциты и гранулоциты, возникает огромное число мелких кластеров, что существенно ухудшает восприятие результатов типологизации. Возможно, что именно по этой причине не всегда удается достоверно отделять дебрис от лимфоцитов. Еще одной проблемой является определение оптимального критерия остановки при построении иерархического дерева или, иными словами, выбор оптимального числа кластеров.

Метод K-средних

Алгоритм *K-means* — популярный метод кластеризации, изобретенный во второй половине прошлого века [24, 25]. Основная идея заключается в том, что на каждой итерации перевычисляется центр масс для каждого кластера, полученного на предыдущем шаге (на первом шаге нужно знать число кластеров и выбрать расположение их центров масс). Затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Моделирование проводили с помощью процедуры *sklearn.cluster.KMeans.* Пример результатов работы процедуры, реализующей метод K-средних, приведен на рис. 3 (см. четвертую сторону обложки).

Среднее время работы программы 3,71 с. Как видно из рис. 3, метод K-средних плохо показал себя на тестовых данных. Ни один из результатов численного моделирования не является удовлетворительным из-за того, что алгоритм разделяет гранулоциты пополам. Возможно, это связано с тем, что метод сходится к локальному минимуму, игнорируя глобальный. Также одним из недостатков алгоритма является то, что итоговые метки классов зависят от первоначального выбора центроидов.

EM-алгоритм

Если *a priori* известно что данные представляют собой смесь нормальных распределений с неизвестными параметрами, то их кластеризация может проводиться с помощью *EM*-алгоритма [26]. Каждая итерация алгоритма состоит из двух шагов. На *E*-шаге для каждого объекта вычисляется вероятность его принадлежности к кластеру. На *M*-шаге вычисляются параметры нормального распределения и решается задача максимального правдоподобия. Затем определяется кластерная принадлежность для каждого объекта. Численное моделирование проводили с помощью процедуры *sklearn.mixture.GaussianMixture.* Результаты работы программного кода, реализующего эту процедуру, приведены на рис. 4 (см. четвертую сторону обложки).

Среднее время работы программы составляет 3,32 с. Как видно из рисунка, этот алгоритм показал себя чуть лучше, чем метод K-средних, однако результат все же не является удовлетворительным. *EM*-алгоритм имеет схожие недостатки с методом K-средних (чувствительность к начальным данным и необходимость задавать число кластеров до начала процесса).

DBSCAN

Последним рассмотрим метод кластеризации, основанный на оценке плотности распределения экспериментальных данных, *DBSCAN (Density-based spatial clustering of applications with noise*) [27]. Этот алгоритм группирует плотно расположенные элементы, помечая как выбросы те точки, которые находятся в областях с малой плотностью. В отличие от метода K-средних и *EM*-алгоритма, *DBSCAN* не требует указывать число кластеров заранее. Предварительно нужно лишь задать параметры, определяющие радиус окрестности и число соседних точек, попадающих в окрестность указанного радиуса. Анализ проводили с помощью процедуры *sklearn.cluster.DBSCAN.* Пример результатов работы алгоритма *DBSCAN* показан на рис. 5 (см. четвертую сторону обложки).

Среднее время работы программы 4,05 с. В целом алгоритм хорошо кластеризовал тестовые данные. Однако у него тоже есть свои недостатки. Во-первых, *DBSCAN* не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию "число соседей" и "радиус". Во-вторых, нет объективного критерия выбора оптимальных параметров, и они подбирались вручную.

Заключение

При численном моделировании процесса кластеризации лейкоцитов по *FS* и *SS* параметрам хуже всего проявил себя метод K-средних. Разделение гранулоцитов на два кластера является ошибочным в принципе. Несущественно лучше результаты кластерного анализа с помощью *EM*-алгоритма. К тому же оба метода предполагают априорное знание числа кластеров.

В целом иерархический алгоритм одиночной связи позволяет получать удовлетворительные результаты, но не более того. Остается открытой проблема завершения процесса кластеризации и определения предпочтительного числа кластеров. Не удается достоверно отделять лимфоциты от мелкого дебриса.

Лучшие результаты были получены при использовании метода *DBSCAN*. Однако при наличии большого объема шумов или большой разницы в плотности распределения лейкоцитов подбор параметров, необходимых для выполнения этого алгоритма, вызывает большие затруднения. Поэтому возникают задачи, связанные с оценкой устойчивости и робастности результатов кластеризации, полученных этим методом.

Основной вывод из проведенных вычислительных экспериментов состоит в том, что необходимо модифицировать существующие алгоритмы так, чтобы они давали хорошие результаты вне зависимости от структуры данных и уровня шумов. Возможным подходом к решению перечисленных выше задач может быть доработка методов *DBSCAN* и одиночной связи с учетом специфики распределения лейкоцитов в осях *FS-SS*.

Список литературы

1. **Хаитов Р. М., Игнатьева Г. А., Сидорович И. Г.** Иммунология: Учебник. М.: Медицина, 2000. 432 с.

2. **Аронова Е. А.** Иммунитет. Теория, философия и эксперимент: Очерки из истории иммунологии XX века. М.: КомКнига, 2006. 160 с.

3. **Смирнова О. А., Степанова Н. В.** Математическая модель колебаний при инфекционном иммунитете // Колебательные процессы в биологических и химических системах: Тр. Второго Всесоюзного симпозиума по колебательным процессам в биологических и химических системах. Пущино-на-Оке: НЦБИ АН СССР, 1971. Т. 2. С. 247—251.

4. **Bell G. I.** Mathematical model of clonal selection and antibody production // J. Theor. Biol. 1970. Vol. 29, N. 2. P. 191—232.

5. **Марчук Г. И.** Математические модели в иммунологии: вычислительные методы и эксперименты. М.: Наука, 1991. 299 с.

6. **Бочаров Г. А., Марчук Г. И.** Прикладные проблемы математического моделирования в иммунологии // Журн. вычисл. математики и матем. физики. 2000. Т. 40, № 12. С. 1905—1920.

7. **Кузнецов С. Р.** Математическая модель иммунного ответа // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2015. № 4. С. 72—87.

8. **Kuznetsov S. R., Kudryavtsev I. V., Orekhov A. V., Polevshchikov A. V., Serebriakova M. K., Shishkin V. I.** A mathematical model for predicting of IGD—CD27 + B lymphocytes levels in donors' blood // Биоинформатика регуляции и структуры геномов и системной биологии. Новосибирск. 2016. С. 166.

9. **Кузнецов С. Р., Лыкосов В. М., Орехов А. В., Шишкин В. И.** Модель пролиферации и дифференцировки неоднородной клеточной популяции // Математическое и компьютерное моделирование в биологии и химии III Международная научная Интернет-конференция. 2014. С. 95—103.

10. **Кузнецов С. Р., Лыкосов В. М., Орехов А. В., Шишкин В. И.** Процесс формирования иммунной памяти Т-лимфоцитов и его зависимость от числа пройденных клетками делений // Устойчивость и процессы управления. Материалы III международной конференции. 2015. С. 487—488.

11. **Зурочка А. В., Хайдуков С. В., Кудрявцев И. В., Черешнев В. А.** Проточная цитометрия в медицине и биологии. 2-е изд. Екатеринбург: УрО РАН, 2014. 574 с.

12. **Балалаева И. В.** Проточная цитофлуориметрия: Учебно-методическое пособие. Нижний Новгород: Нижегородский госуниверситет, 2014. 75 с.

13. **Основы** доказательной медицины: Учебное пособие для системы послевузовского и дополнительного профессионального образования врачей / Под общей редакцией академика РАМН, профессора Р. Г. Оганова. М.: Силицея-Полиграф, 2010. 136 с.

14. **Pedersen N. W.** Automated Analysis of Flow Cytometry Data to Reduce Inter-Lab Variation in the Detection of Major Histocompatibility Complex Multimer-Binding T Cells // Front Immunol 8: 2017. p. 858.

15. **Daneau G.** CD4 results with a bias larger than hundred cells per microliter can have a significant impact on the clinical decision during treatment initiation of HIV patients // Cytometry B Clin Cytom 92(6): 2017. P. 476—484.

16. **Qian Yu.** FlowGate: towards extensible and scalable web-based flow cytometry data analysis. XSEDE, 2015.

17. **Omana-Zapata I.** Accurate and reproducible enumeration of T-, B-, and NK lymphocytes using the BD FACSLyric 10-color system: A multisite clinical evaluation // PLoS One 14(1): 2019. e0211207.

18. **Орехов А. В.** Марковский момент остановки агломеративного процесса кластеризации в евклидовом пространстве // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2019. Т. 15, Вып. 1. С. 76—92.

19. **Rousseeuw P. J.** "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics.

20. **Amorim R. C., Hennig C. (2015).** "Recovering the number of clusters in data sets with noise features using feature rescaling factors" // Information Sciences. 324: 126—145.

21. **Everitt B. S.** Cluster Analysis. Chichester, West Sussex, UK: John Wiley & Sons Ltd, 2011. 330 p.

22. **Hartigan J. A.** Clustering algorithms. New York, London, Sydney, Toronto: John Wiley & Sons Inc. Press, 1975. 351 p.

23. **Aldenderfer M. S., Blashfield R. K.** Cluster analysis. Newburg Park, Sage Publications Inc. Press, 1984. 88 p.

24. **Steinhaus H.** Sur la division des corps materiels en parties // Bull. Acad. Polon. Sci. C1. III. 1956. Vol. IV. P. 801—804.

25. **Lloyd S.** Least squares quantization in PCM // IEEE Transactions on Information Theory. 1982. Vol. 28, Iss. 2. P. 129—137. doi:10.1109/TIT.1982.1056489.

26. **Dempster A. P., Laird N. M., Rubin D. B.** Maximum Likelihood from Incomplete Data via the EM Algorithm // Journal of the Royal Statistical Society, Series B. 1977. Vol. 39 (1). P. 1—38.

27. **Ester M., Kriegel H.-P., Sander J., Xu. X.** A density-based algorithm for discovering clusters in large spatial databases with noise // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). Evangelos Simoudis, Jiawei Han, Usama M. Fayyad. AAAI Press, 1996. P. 226—231.

A. I. Lepsky, alexlep97@gmail.com,
Saint Petersburg State University, Saint Petersburg, ..., Russian Federation

Comparative Analysis of Leukocyte Clustering Algorithms According to FS and SS Parameters in a Cytofluorimetric Blood Test

In laboratory studies in biology and medicine, obtaining formal rules for evaluating the numerical values of empirical data is of great practical importance. One of the unsolved problems in the cytometric blood test is the automatic typology of leukocytes. A possible approach, in this case, could be the use of cluster analysis methods. However, with the clustering of white blood cells, many unresolved issues remain. The article explores various clustering algorithms. When conducting numerical experiments, it was shown that hierarchical methods and the K-means method do not give positive results. The DBSCAN method is the most promising for further study. A program code written in Python was created to conduct numerical experiments.

Keywords: *cluster analysis, flow cytometry, machine learning*

DOI: 10.17587/it.26.56-61

References

1. **Haitov R. M., Ignat'eva G. A., Sidorovich I. G.** Immunologiya: Uchebnik, Moscow, Medicina, 2000, 432 p. (in Russian).

2. **Aronova E. A.** Immunity. Theory, Philosophy, and Experiment: Essays on the History of Immunology of the 20th Century, Moscow, KomKniga, 2006, 160 p. (in Russian).

3. **Smirnova O. A., Stepanova N. V.** *Kolebatel'nye processy v biologicheskih i himicheskih sistemah: Trudy Vtorogo Vsesoyuznogo simpoziuma po kolebatel'nym processam v biologicheskih i himicheskih sistemah*. Pushchino-na-Oke: NCBI AN SSSR, 1971, vol. 2, pp. 247—251 (in Russian).

4. **Bell G. I.** *J. Theor. Biol*., 1970, vol. 29, no. 2, pp. 191—232.

5. **Marchuk G. I.** Mathematical models in immunology: computational methods and experiments, Moscow, Nauka, 1991, 299 p. (in Russian).

6. **Bocharov G. A., Marchuk G. I.** *ZHurn. vychisl. matematiki i matem. Fiziki*, 2000, vol. 40, no. 12, pp. 1905—1920 (in Russian).

7. **Kuznecov S. R.** *Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya\linebreak matematika. Informatika. Processy upravleniya*, 2015, no. 4, pp. 72—87 (in Russian).

8. **Kuznetsov S. R., Kudryavtsev I. V., Orekhov A. V., Polevshchikov A. V., Serebriakova M. K., Shishkin V. I.** *Bioinformatika regulyacii i struktury genomov i sistemnoj biologii*, Novosibirsk, 2016, p. 166.

9. **Kuznecov S. R., Lykosov V. M., Orekhov A. V., Shishkin V. I.** *Matematicheskoe i komp'yuternoe modelirovanie v biologii i himii III Mezhdunarodnaya nauchnaya Internet-konferenciya*, 2014, pp. 95—103 (in Russian).

10. **Kuznecov S. R., Lykosov V. M., Orekhov A. V., Shishkin V. I.** *Ustojchivost' i processy upravleniya. Materialy III mezhdunarodnoj konferencii,* 2015, pp. 487—488 (in Russian).

11. **Zurochka A. V., Hajdukov S. V., Kudryavcev I. V., Chereshnev V. A.** Flow cytometry in medicine and biology, Ekaterinburg: UrO RAN, 2014, 574 p. (in Russian).

12. **Balalaeva I.V** Flow cytofluorimetry: a teaching aid, Nizhnij Novgorod, Nizhegorodskij gosuniversitet, 2014, 75 p. (in Russian).

13. **Oganov R. G.** ed. The basics of evidence-based medicine. Textbook for postgraduate and continuing professional education of doctors, Moscow, Siliceya-Poligraf, 2010, 136 p. (in Russian).

14. **Pedersen N. W.** Automated Analysis of Flow Cytometry Data to Reduce Inter-Lab Variation in the Detection of Major Histocompatibility Complex Multimer-Binding T Cells, *Front Immunol* 8: 2017, p. 858.

15. **Daneau G.** CD4 results with a bias larger than hundred cells per microliter can have a significant impact on the clinical decision during treatment initiation of HIV patients, *Cytometry B Clin Cytom* 92(6): 2017, pp. 476—484.

16. **Qian Yu.** FlowGate: towards extensible and scalable web-based flow cytometry data analysis, XSEDE, 2015.

17. **Omana-Zapata I.** Accurate and reproducible enumeration of T-, B-, and NK lymphocytes using the BD FACSLyric 10-color system: A multisite clinical evaluation, *PLoS One* 14(1): 2019, e0211207.

18. **Orekhov A. V.** *Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya matematika. Informatika. Processy upravleniya*, 2019, vol. 15, iss. 1, pp. 76—92 (in Russian).

19. **Rousseeuw P. J.** Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis, Computational and Applied Mathematics.

20. **Amorim R. C., Hennig C.** Recovering the number of clusters in data sets with noise features using feature rescaling factors". Information Sciences, 2015, 324: 126—145.

21. **Everitt B. S.** Cluster Analysis. Chichester, West Sussex, UK, John Wiley & Sons Ltd, 2011, 330 p.

22. **Hartigan J. A.** Clustering algorithms. New York, London, Sydney, Toronto, John Wiley & Sons Inc. Press, 1975, 351 p.

23. **Aldenderfer M. S., Blashfield R. K.** Cluster analysis. Newburg Park, Sage Publications Inc. Press, 1984, 88 p.

24. **Steinhaus H.** Sur la division des corps materiels en parties, *Bull. Acad. Polon. Sci.* C1. III. 1956, vol. IV, pp. 801-804.

25. **Lloyd S.** Least squares quantization in PCM, *IEEE Transactions on Information Theory*, 1982, vol. 28, iss. 2, pp. 129—137, doi:10.1109/TIT.1982.1056489

26. **Dempster A. P., Laird N. M., Rubin D. B.** Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society, Series B,* 1977, vol. 39 (1), pp. 1—38.

27. **Ester M., Kriegel H.-P., Sander J., Xu. X.** A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)* / Evangelos Simoudis, Jiawei Han, Usama M. Fayyad. AAAI Press, 1996, pp. 226—231.

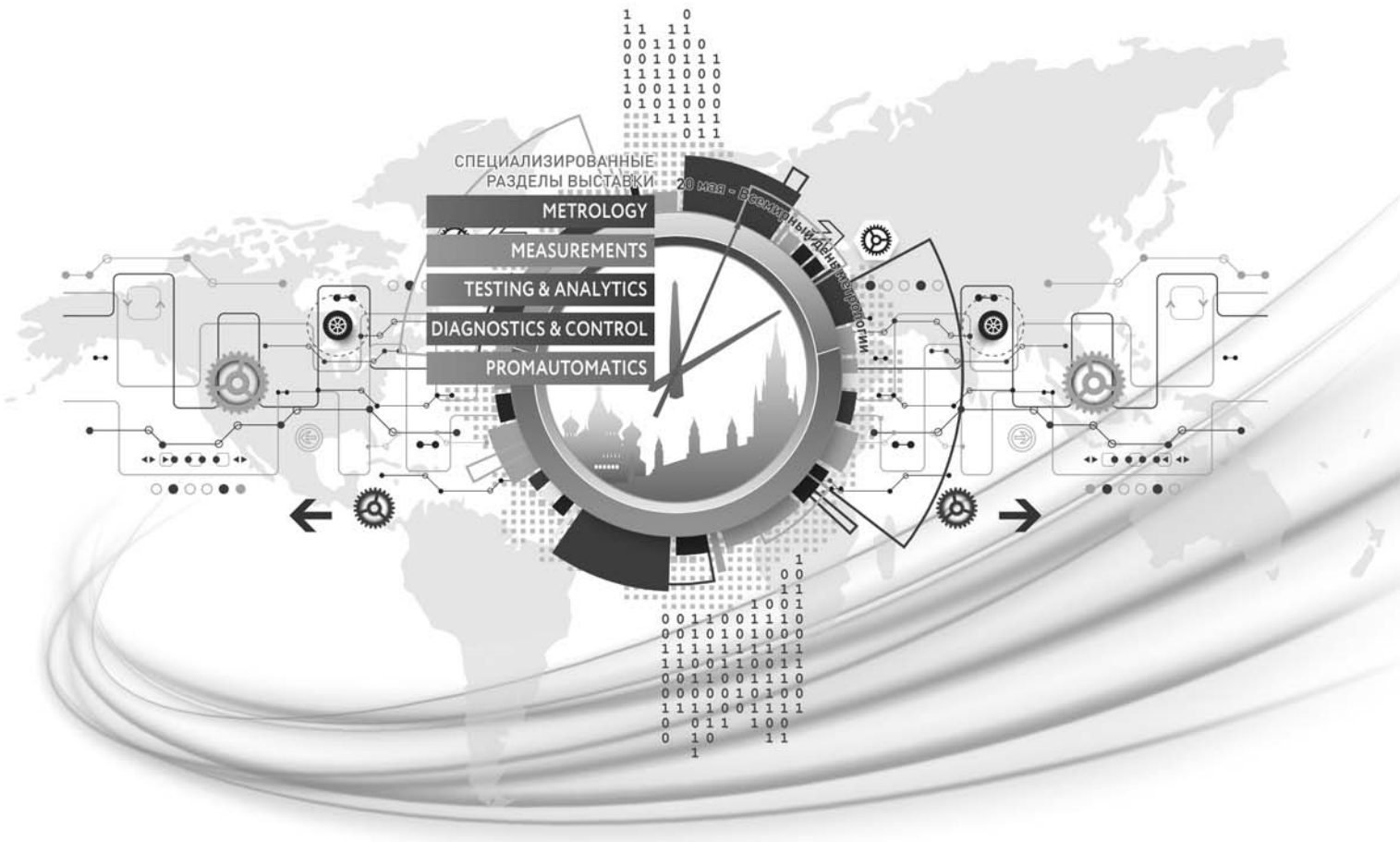
16-й МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ
ИННОВАЦИОННЫЙ ФОРУМ И ВЫСТАВКА

ТОЧНЫЕ ИЗМЕРЕНИЯ –
ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

MetrolExpo'2020

Москва, 2-4 июня

ВДНХ, павильон 75



ТЕМАТИЧЕСКИЕ РАЗДЕЛЫ ВЫСТАВКИ:

МЕТРОЛОГИЯ
METROLOGY

ИЗМЕРЕНИЯ
MEASUREMENTS

ИСПЫТАНИЯ и АНАЛИТИКА
TESTING & ANALYTICS

ДИАГНОСТИКА и КОНТРОЛЬ
DIAGNOSTICS & CONTROL

АВТОМАТИЗАЦИЯ
PROMAUTOMATICS

Устроитель:
Выставочная компания
ВЭСТСТРОЙ ЭКСПО
+7 (495) 937-40-23
metrol@expoprom.ru

ПРИГЛАШАЕМ ПРИНЯТЬ УЧАСТИЕ

www.metrol.expoprom.ru

Конференции по информационным технологиям в 2020 году

Конференция

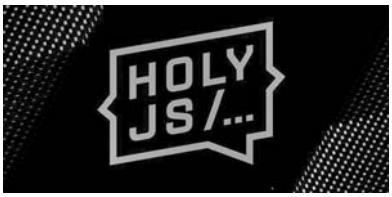
Информационные технологии в промышленности 2020

Даты проведения: 19.03.2020—20.03.2020 г.
Место проведения: Россия, Москва

Конференция

DotNext Piter 2020

Даты проведения: 06.04.2020—07.04.2020 г.
Место проведения: Россия, Санкт-Петербург
Основные темы конференции:
- настоящее и будущее платформы.NET;
- оптимизация производительности;
- внутреннее устройство платформ;
- архитектура и паттерны проектирования;
- нетривиальные задачи и best practices.



Конференция

HolyJS Piter 2020

Даты проведения: 10.04.2020—11.04.2020 г.
Место проведения: Россия, Санкт-Петербург
Большая конференция для JavaScript-разработчиков.

Конференция

OFFZONE 2020

Даты проведения: 16.04.2020—17.04.2020 г.
Место проведения: Россия, Москва
Сердце конференции OFFZONE — качественный технический контент и презентация практических исследований в области кибербезопасности.

Конференция

Blockchain Life 2020

Даты проведения: 22.04.2020—23.04.2020 г.
Место проведения: Россия, Москва
Международный форум по блокчейну, криптовалюте и майнингу

Конференция

C++ Russia 2020

Даты проведения: 27.04.2020—28.04.2020 г.
Место проведения: Россия, Москва
Конференция посвящена проблемам C++: concurrency, производительность, архитектура и инфраструктурные решения.

Международный конгресс по кибербезопасности 2020

Даты проведения: 09.07.2020—10.07.2020 г.
Место проведения: Россия, Москва
Мероприятие посвящено обсуждению вопросов обеспечения кибербезопасности в условиях общемировой дигитализации.

Специализированная выставка

Безопасность. IT-технологии. Коммуникации. Связь 2020

Даты проведения: 21.05.2020—23.05.2020 г.
Место проведения: Россия, Челябинск

Тематические направления выставки:

- IT-системы и оборудование
- IT-услуги, консалтинг, интернет-технологии
- Мобильная и спутниковая связь, IP-телефония
- Сети передачи данных мобильные сети
- Программное обеспечение
- Системы и технические средства видеонаблюдения
- Программы по обеспечению комплексной безопасности
- Методы, технологии и оборудование для обеспечения безопасности
- Системы защиты информации и управления данными
- Телекоммуникационные технологии безопасности

12-й Международный форум

IT-форум — Югра 2020

Даты проведения: 16.06.2020—17.06.2020 г.
Место проведения: Россия, Ханты-Мансийск

Тематика форума в области информационных технологий

- Электронные регионы, электронные муниципалитеты
- ИТ-парки; ИТ-бизнес-инкубаторы
- Информационно-коммуникационные технологии
- Электронный документооборот в органах государственной власти
- Информационных технологий для взаимодействия государства с бизнесом
- Использование инфокоммуникаций в социальной сфере
- Системы идентификации пользователя, системы защиты информации
- Телекоммуникации, как средство человеческого общения
- Средства развлечения, использование инфокоммуникационных систем доступа
- Защита персональных данных

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**
E-mail: it@novtex.ru

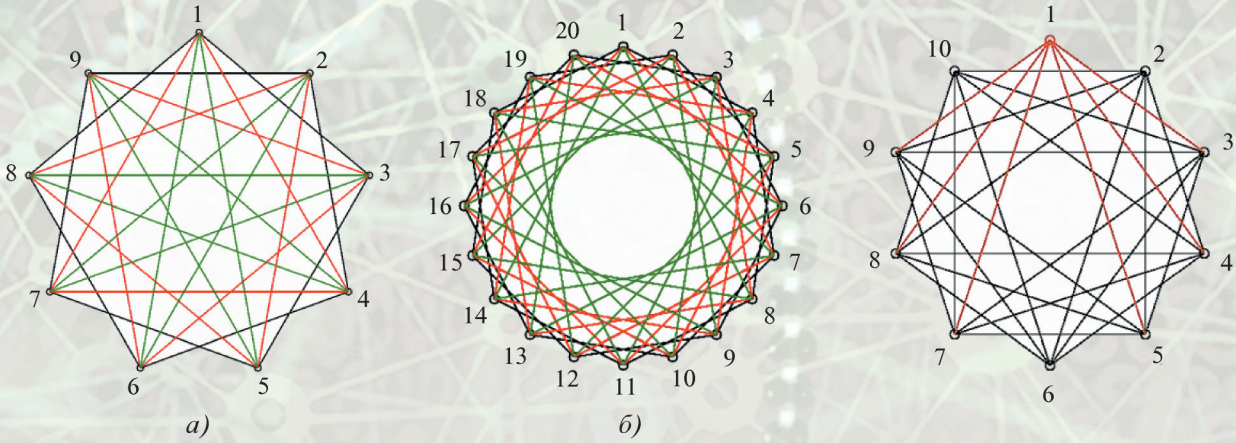
Технический редактор *Е. В. Конова*.
Корректор *Н. В. Яшина*.
Сдано в набор 07.11.2019. Подписано в печать 23.12.2019. Формат 60×88 1/8. Бумага офсетная.
Усл. печ. л. 8,86. Заказ IT120. Цена договорная.

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

Рисунки к статье А. Ю. Романова, М. В. Сидоренко, Э. А. Монаховой

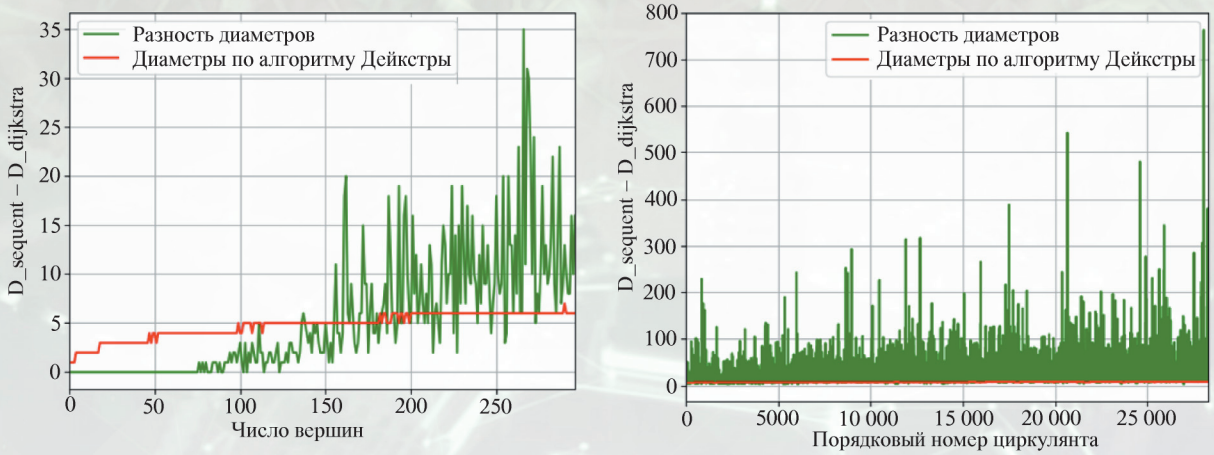
«МАРШРУТИЗАЦИЯ В СЕТЯХ-НА-КРИСТАЛЛЕ С ТОПОЛОГИЕЙ ТРЕХМЕРНЫЙ ЦИРКУЛЯНТ»



а) *б)*

Рис. 1. Циркулянты с тремя образующими: *а* – *C*(9; 2, 3, 4); *б* – *C*(20; 3, 5, 7)

Рис. 2. Единичные маршруты из одной вершины в циркулянте *C*(10; 2, 3, 4)



а) *б)*

Рис. 5. График разностей максимальных расстояний, рассчитанных с помощью алгоритмов «Sequent» и Дейкстры: *а* – тест 2; *б* – тест 3

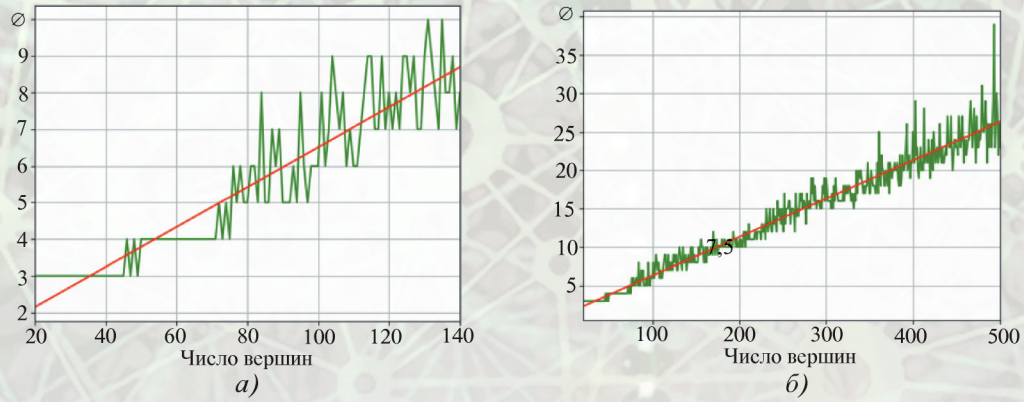
а) *б)*

Рис. 6. Зависимость времени выполнения алгоритма от числа вершин в циркулянтах типа: *а* – *C*(*N*; 2, 3, 7); *б* – *C*(*N*; 10, 17, 39)

Рис. 7. Зависимость максимального расстояния от числа вершин для циркулянтов типа *C*(*N*; 2, 3, 7) при изменении *N* от 20 до 220

Рисунки к статье А. Ю. Романова, М. В. Сидоренко, Э. А. Монаховой

«МАРШРУТИЗАЦИЯ В СЕТЯХ-НА-КРИСТАЛЛЕ С ТОПОЛОГИЕЙ ТРЕХМЕРНЫЙ ЦИРКУЛЯНТ»



а) *б)*

Рис. 8. Зависимость максимального расстояния циркулянта, рассчитанного по разработанному алгоритму, от числа вершин для циркулянтов типа *C*(*N*; 4, 7, 13) при изменении *N*: *а* – от 20 до 140; *б* – от 20 до 500

Рисунки к статье А. И. Лепского

«СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ЛЕЙКОЦИТОВ ПО FS И SS ПАРАМЕТРАМ ПРИ ЦИТОФЛУОРИМЕТРИЧЕСКОМ ИССЛЕДОВАНИИ КРОВИ»

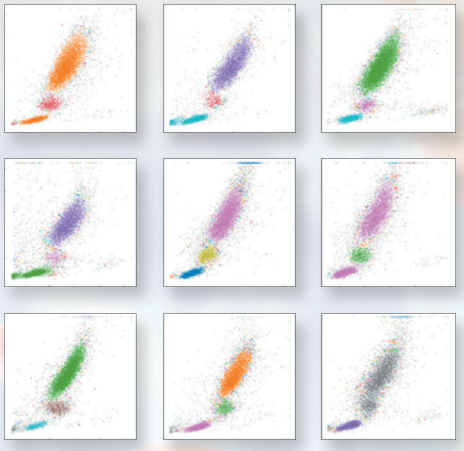


Рис. 2. Результаты численного моделирования процесса кластеризации лейкоцитов методом одиночной связи

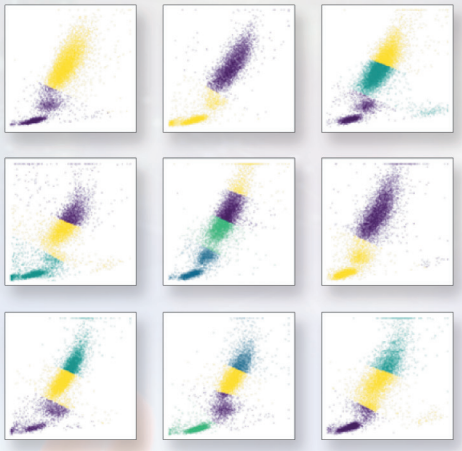


Рис. 3. Результаты численного моделирования методом К-средних

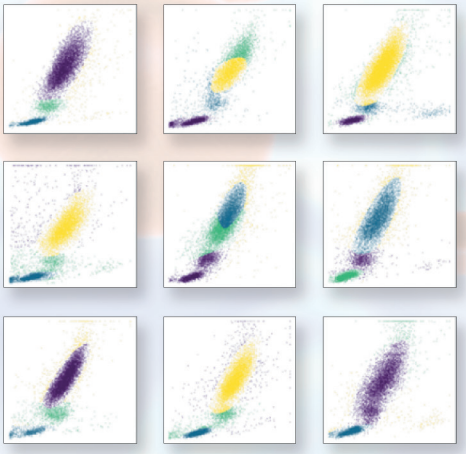


Рис. 4. Результаты кластеризации с помощью *EM*-алгоритма

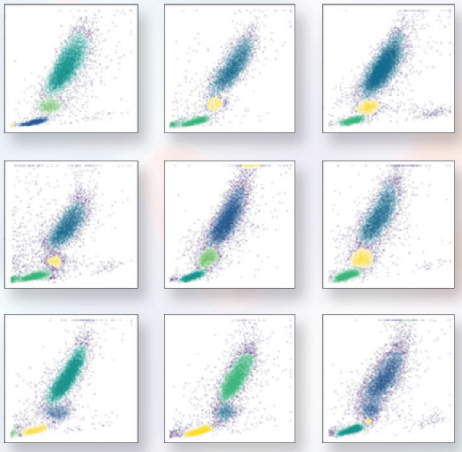


Рис. 5. Результаты кластеризации лейкоцитов методом *DBSCAN*

ISSN 1684-6400. Информационные технологии. 2020. Том 26. № 1. 1–64. Индекс: 72656 (Объединенный каталог «Пресса России»)