Виктор А. Шурыгин[1], Игорь М. Ядыкин[2]
Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, Москва, 115409, Россия
[1]e-mail: vic-54@mail.ru, https://orcid.org/0000-0003-2739-9560
[2]e-mail: IMYadykin@mephi.ru, https://orcid.org/0000-0003-3952-5288

УНИВЕРСАЛЬНОЕ КОДИРОВАНИЕ ТРОЙКАМИ ДЛЯ СЖАТИЯ И ЗАЩИТЫ ДВОИЧНЫХ ДАННЫХ

DOI: http://dx.doi.org/10.26583/bit.2021.1.01

Аннотация. Статья посвящена актуальной проблеме сжатия обрабатываемых и передаваемых данных, хранения и защиты данных. Приводится анализ и особенности методов сжатия данных. Рассмотрен метод универсального кодирования тройками двоичных наборов (КТ). Метод предназначен для сжатия двоичных данных без потерь в условиях неизвестной статистики источника сообщений. Метод основан на разбиении исходной последовательности двоичных данных на блоки разрядностью *n*. Каждому блоку на основании анализа содержимого *n*–блока ставится в соответствие три параметра: количество единиц в блоке, сумма номеров позиций единиц, номер данной конкретной комбинации в соответствующем префиксном классе. Рассмотрены подходы к снижению трудоемкости процедур вычисления параметров КТ. Приводится табличный алгоритм вычисления коэффициентов и алгоритм заполнения строк таблиц с применением рекуррентного соотношения для элементов множеств, оценка объемов памяти для хранения таблиц и способы сокращения емкости памяти. Рассмотрена зависимость разрядности кодовых слов КТ от разрядности *n*-блоков. Анализируются подходы и особенности применения кодировании тройками для защиты информации.

Ключевые слова: сжатие данных, кодирование, хранение данных, защита данных, рекуррентное отношение, сжатие без потерь.

Для цитирования: ШУРЫГИН, Виктор А.; ЯДЫКИН, Игорь М. УНИВЕРСАЛЬНОЕ КОДИРОВАНИЕ ТРОЙКАМИ ДЛЯ СЖАТИЯ И ЗАЩИТЫ ДВОИЧНЫХ ДАННЫХ. Безопасность информационных технологий, [S.l.], v. 28, n. 1, p. 6–18, jan. 2021. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1315>. Дата доступа: 14 jan. 2021. DOI: http://dx.doi.org/10.26583/bit.2021.1.01

Victor A. Shurygin[1], Igor M. Yadykin[2]
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashirskoe shosse, 31, Moscow, 115409, Russia
[1]e-mail: vic-54@mail.ru, https://orcid.org/ 0000-0003-2739-9560
[2]e-mail: IMYadykin@mephi.ru, https://orcid.org/0000-0003-3952-5288

Universal triple encoding for compression and protecting binary data

DOI: http://dx.doi.org/10.26583/bit.2021.1.01

Abstract. The paper is devoted to the actual problem of compression of processed and transmitted data, as well as of the data storage and protection. The analysis and features of data compression methods are presented. The method of universal coding by triples of binary sets (CT) is considered. The method is intended for lossless compression of binary data under conditions of unknown message source statistics. The method is based on dividing the original sequence of binary data into blocks of length *n*. Based on the analysis of the contents of the n–block, three parameters are assigned to each block: the number of units in the block, the sum of the unit position numbers, and the number of this particular combination in the corresponding prefix class. Approaches to reducing the complexity of procedures for calculating the parameters of CT are considered. A tabular algorithm for calculating coefficients and an algorithm for filling table rows using a recursive relation for elements of sets and an estimate of the amount of memory for storing tables as well as the ways to reduce the memory capacity are presented. The dependence of the

bit width of the QT codewords on the bit width of *n*-blocks is considered. The approaches and features of the use of coding by triplets for information protection are analyzed.
Keywords: data compression, encoding, data storage, data protection, recurrence relation, lossless compression.
For citation: SHURYGIN, Victor A.; YADYKIN, Igor M. Universal triple encoding for compression and protecting binary data. IT Security (Russia), [S.l.], v. 28, n. 1, p. 6–18, jan. 2021. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/1315>. Date accessed: 14 jan. 2021. DOI: http://dx.doi.org/10.26583/bit.2021.1.01

Введение

В настоящее время наблюдается лавинообразный рост объемов передаваемых, хранимых и обрабатываемых двоичных данных. Фактически не осталось областей интеллектуальной деятельности человека, где это не ощущалось бы. Это эксперименты в ядерной физике, космических исследованиях, ИТ областях и пр. Особенно следует отметить специфику бортовых применений, где идет постоянная борьба за каждый грамм, миллиметр, литр. Для экономии объемов памяти и более эффективного использования информационных каналов используется сжатие данных. Но и здесь возникают специфические проблемы. Дополнительные трудности определяются, тем, что, как правило, статистика по данным исследуемых объектов неизвестна или известна неполностью. Понятно, что здесь невозможно «в лоб» применить принцип – более вероятному сообщению ставим в соответствие более короткое кодовое слово и за счет этого выигрываем в объеме. Особенно остро эти проблемы дают себя знать в случае необходимости использования квазиобратимого сжатия, или сжатия без потерь. В силу этого появились различные подходы к устранению статистической избыточности из потока сообщений.

1. Краткий обзор методов сжатия

В [1] авторами приведена классификация подходов к методам сжатия информации. Все методы делятся на две большие группы – сжатие допускающее потерю части информации и сжатие без потерь. Например, при передаче аудиофайлов можно «угрубить» – допустить ухудшение качества звука незаметное подавляющему большинству людей из-за их антропологических свойств и за счет этого существенно уменьшить нагрузку на каналы передачи данных и сэкономить память [2–3]. Однако методы сжатия с потерями обладают недостатками: применима не для всех случаев графической информации, например, при исследовании медицинских снимков потери могут быть недоступны глазу, но могут быть доступны для анализатора, второй недостаток – накопление погрешности при повторной компрессии и декомпрессии [4–6].

Кодирование без потерь может применяться для сжатия любой информации, поскольку обеспечивает точное восстановление данных после кодирования и декодирования.[3] Сжатие без потерь основано на простом принципе преобразования данных из одной группы символов в другую, более компактную. Когда потери недопустимы, единственно возможный подход – из последовательности символов удаляется статистическая избыточность и объем сообщения приближается к энтропии.

Если статистика известна, например, вероятность появления единиц в бинарной последовательности значительно меньше, чем нулей, то более вероятным сообщениям ставим в соответствие более короткие кодовые слова – удаляем из потока длинные последовательности одинаковых символов и т.д. Здесь можно отметить адресно-позиционное кодирование (АПК) [7], кодирование длин серий (КДС) [7–8], коды

Хаффмана [7, 9], на основе которых строятся архиваторы [2, 8, 10], исключение избыточных нулевых групп [11, 12] и т.д.

В 1967 г. Б.М. Фитингофом, на основе работ А.Н. Колмогорова, была рассмотрена возможность создания кодов устраняющих статистическую избыточность из потока символов без знания статистики источника их порождающего [13–15]. В [13] было строго математически доказана обоснованность такого подхода.

Данное направление получило название – универсальное кодирование. Суть в том, что исходная последовательность разбивается на блоки длиной n, затем применяется кодирование, и статистическая избыточность в потоке стремится к нулю при стремлении n к бесконечности. В работах [2, 6, 13] были рассмотрены различные аспекты такого подхода. Следует отметить основную трудность, препятствующую практическому применению универсального кодирования – это высокая трудоемкость.

В теории информации трудоемкость оценивалась двумя параметрами – время и объем памяти, требуемые для реализации кодирования. Б.М. Фитингоф определил рост объема памяти с ростом n, как экспоненту. В то время это было немыслимо. Появились работы направленные на уменьшение объема памяти за счет построения алгоритмов нумерации элементов кодирования, т.е. уход от табличной реализации [16]. Понятно, что это вело к резкому росту времени реализации кодирования.

В настоящее время состояние элементной базы кардинально изменилось, что делает возможным перейти к реальной реализации универсального кодирования. Одному из возможных методов кодирования и его реализации посвящена эта статья.

2. Метод кодирования тройками

В [17] описан разработанный авторами метод универсального кодирования тройками. В данном методе входные двоичные последовательности размерности N разбиваются на двоичные блоки длиной n бит (n–блоки) (где $n< N$), для каждого n–блока вычисляются три параметра. При этом если разрядность параметров меньше чем 2^n, то осуществляется сжатие данных.

На рис. 1 для n=5 приведены возможные двоичные числа разрядностью n бит, всего 2^n=32 элементов, расположенные в порядке возрастания. Для множества N введем два подмножества M_k и L_S:

– M_k элементами подмножества являются все элементы множества N, содержащие k единиц $(0 \le k \le n)$.

– L_S элементами подмножества являются все элементы множества N, сумма номеров позиций единиц которых равна S $(0 \le S \le n(n+1)/2)$.

На рис. 1 показаны подмножества M_k и L_S для n=5. Цифрами 1, …, 5 заданы номера бит (разрядов) в n–блоке.

Введем подмножество $R_{k,\,S}$, как пересечение подмножеств M_k и L_S. Обозначим $r(n,\,k,\,s)$ – количество элементов множества $R_{k,\,S}$. Поставим в соответствие каждому элементу подмножества $R_{k,\,S}$ номер $b(n,\,k,\,s)$, причем $0 \le b(n,\,k,\,s) \le r(n,\,k,\,s) - 1$.

Авторами в [18] проведено исследование эффективности универсального кодирования в зависимости от длины n–блока. Получены зависимости избыточности кодирования и коэффициента сжатия от длины n–блока.

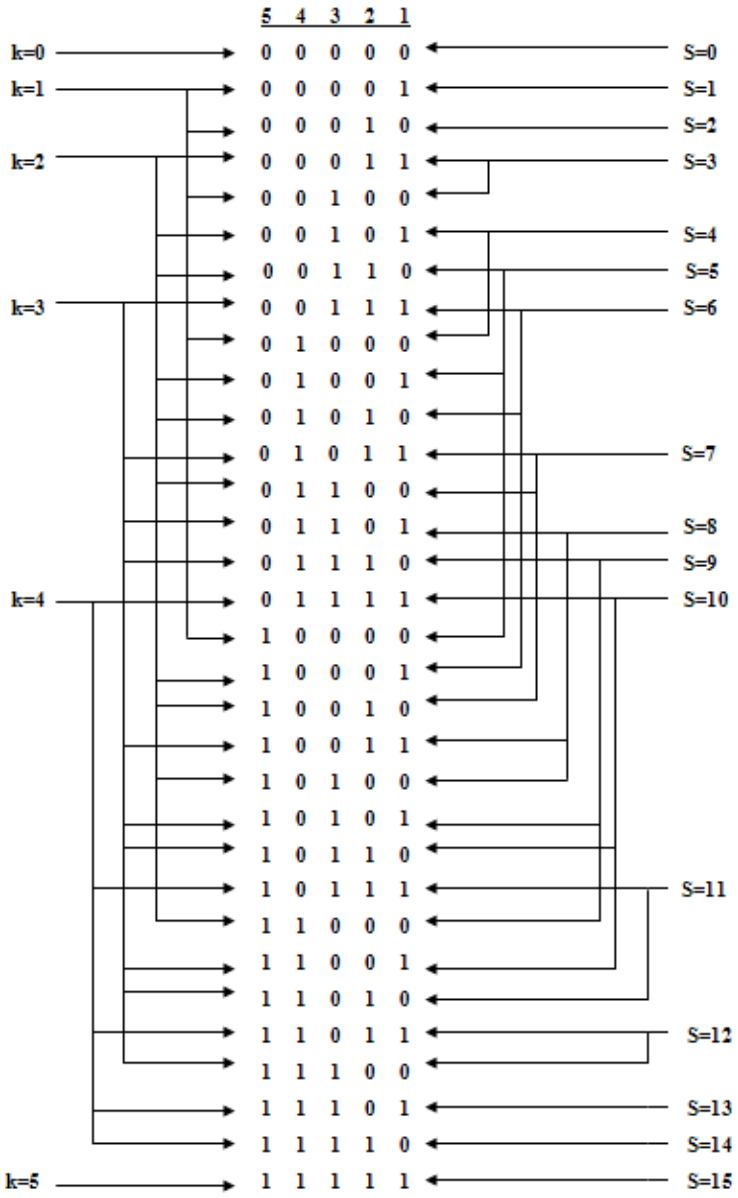

Рис. 1. Элементы множества N для n=5
(Fig. 1. Elements of the set N for n = 5)

Определим кодовое слово w соответствующее n–блоку, как упорядоченную тройку двоичных наборов $(k, s, b(n, k, s))$. Разрядность кодового слова w (в битах) будет равна:

$$Z =]log_2 (n+1)[+]log_2 (n (n+1)/2)+1[+]log_2 r(n, k, s)[, \quad (1)$$

где] [большее целое; первое слагаемое количество разрядов для задания максимального числа единиц в n–блоке; второе слагаемое – для задания максимального числа суммы всех номеров позиций в n–блоке; третье слагаемое – для задания максимального числа параметра $r(n, k, s)$ пересечения подмножеств M_k и L_s.

Введем подмножество W, элементами которого являются все кодовые слова w. Из формирования подмножества W следует, что W – префиксное множество, а, следовательно, между n-блоками и кодовыми словами w существует взаимно однозначное соответствие.

Кодирование n-блоков двоичных наборов (k, s, $b(n, k, s)$) авторами в [17] названо как **кодирование тройками (КТ).**

В литературе по комбинаторике [19–21] не было выявлено соотношений для вычисления значений коэффициентов $r(n, k, s)$. В [17] авторами были сформулированы и доказаны теоремы для вычисления коэффициентов r (n, k, s) и выведено рекуррентное соотношение:

$$r(n, k, s) = r(n–1, k, s)+ r(n–1, k–1, s–n). \quad (2)$$

Далее для определения номера $b(n, k, s)$ соответствующего каждому элементу подмножества $R_{k,S}$ был разработан алгоритм нумерации.

В n–блоке фиксируем номер k–ой единицы i_k. При этом остальные номера от 1 до (i_k–1) будут размещаться в (i_k–1) разрядной сетке и количество всех возможных комбинаций k номеров с суммой s, будет равно $r(i_k–1, k, s)$. Аналогично можно получить значения для всех номеров до i_1 включительно. В результате для каждого i_j получено соответствующее значение $r((i_j–1), j, (s – i_k – i_{k-1} –…– i_{j+1}))$. На основании исследований и доказательств получено соотношение для вычисления номера:

$$b(n, k, s)=r((i_k –1), k, (i_k + i_{k-1}+...+ i_1)) + r((i_{k-1} –1), (k\text{-}1), (i_{k-1} + i_{k-2} ...+i_1))+ ... +$$

$$+ r((i_2 – 1), 2, (i_2 + i_1)) = (\sum_{j=2}^{k} r((i_j – 1)), j, (\sum_{m=1}^{j} i_m)). \quad (3)$$

При этом любой номер b (n, k, s), вычисленный по формуле (3) удовлетворяет неравенству: $0 \le b(n, k, s) \le r(n, k, s) – 1$.

3. Табличный алгоритм вычисления коэффициентов

Трудоемкость определения коэффициентов КТ определяется вычислением $r(n, k, s)$. На рис. 2а приведена полная таблица 1 значений коэффициентов $r(n, k, s)$ для n=7. Например, коэффициенту $r(7, 2, 8)=3$ при двух единицах $k=2$ и сумме позиций единиц $s=8$ соответствует три следующих последовательности в n–блоке ($n=7$) – 0010100, 0100010, 1000001. При $k=1$ для каждого значения s (от 1 до 7) соответствует только по одной последовательности содержащей одну единицу. При $k=7$ возможна только одна последовательность в n–блоке ($n=7$) содержащая все единицы 1111111, при этом единицам соответствует сумма всех позиций в n–блоке ($n=7$) – $s=28$.

Оценим объем таблицы. Таблица (рис. 1b) содержит k строк ($0 \le k \le$ n+1) и s столбцов ($0 \le s \le$ (n*(n+1)/2 + 1)). Общее количество ячеек памяти для элементов таблицы составляет V:

$$V = (n+1) (n (n+1)/2 + 1)) = n (n+1)^2/2 + n + 1 \quad (4)$$

Их таблицы видно, что первая строка $k=1$ симметрична шестой строке $k=6$, вторая строка $k=1$ симметрична пятой строке $k=5$, третья строка $k=3$ симметрична четвертой строке $k=4$, при сдвигах в столбцах соответственно на 20, 12 и 4 позиции.

Из анализа коэффициентов $r(n, k, s)$ было выявлено, что коэффициенты обладают свойством симметрии – для любых значений n, k, s выполняются условия:

$$r(n, k, s) = r(n, (n\text{-}k), (n\text{-}(n+1))/2 – s)), \quad (5)$$

$$r(n, k, s) = r(n, k, (k\ (n+1)) – s)). \quad (6)$$

k \ S	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	1																												
1		1	1	1	1	1	1	1																				
2				1	1	2	2	3	3	3	2	2	1	1															
3							1	1	2	3	4	4	5	4	4	3	2	1	1										
4											1	1	2	3	4	4	5	4	4	3	2	1	1						
5																1	1	2	2	3	3	3	2	2	1	1			
6																						1	1	1	1	1	1	1	
7																													1

a)

k \ S	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1		1	1	1	1								
2				1	1	2	2	3	3				
3							1	1	2	3	4	4	5

b)

k \ S	1	2	3	4	5	6	7
1	1	1	1	1			
2	1	1	2	2	3	3	
3	1	1	2	3	4	4	5

c)

Рис. 2. Таблицы коэффициентов r(n, k, s) при n=7
а) полная таблица, в) симметричная таблица, с) минимальная таблица
(Fig. 2. Tables of coefficients r(n, k, s) for n = 7
a) complete table, b) symmetric table, c) minimal table)

В соответствии с соотношениями (5), (6) полная таблица (рис. 2a) может быть преобразована к виду таблицы с учетом симметрии коэффициентов (рис. 2b). При этом при обращении к таблице для определения коэффициентов *r*, для произвольных *k* и *s*, проводится сравнение с соответствующими максимальными значениями:

$$k_{макс} =]n/2[, \qquad s_{макс} =](k_{макс}\ (n+1))/2[\qquad (7)$$

Если *k* и *s* будут меньше или равны, то непосредственно обращаемся к таблице. Если больше, то преобразуем по соотношениям (5) и (6) и затем обращаемся к таблице.

Для таблицы с учетом симметрии (рис. 2b) общее количество ячеек памяти составляет:

$$V = ([n/2]+1)\ ([([n/2]\ (n+1))/2] + 1 \qquad (8)$$

Для нечетных *n* соотношение (8) преобразуется к виду:

$$V = ((n+1)\ (n^2+3))/8$$

Для четных *n* делящихся на 4 соотношение (8) преобразуется к виду:

$$V = ((n+1)\ (n^2+n+4))/8$$

Для четных *n* не делящихся на 4 соотношение (8) преобразуется к виду:

$$V = ((n+1)\ (n^2+n+2))/8$$

Из проведенных преобразований следует, что объем таблицы с учетом симметрии (рис. 2b) сокращается более чем в 4 раза по сравнению с объемом (4) для полной таблицы (рис. 2a).

В таблице с учетом симметрии (рис. 2б) можно заметить «пустые» клетки соответствующие таким значениям k и s для которых отсутствуют двоичные комбинации в n–блоках. Например, невозможно в 7 разрядах так расположить 2 единицы (k=2), чтобы сумма их позиций s была равна 2 или более 15 и т.п. Другие «пустые» клетки, например, k=2 и s=9 или k=1 и s=5 (в таблице заштрихованы) возникают за счет того, что перед обращением к таблице коэффициентов, сначала осуществляется преобразование k и s в соответствии с (5) и (6). Таким образом, к ячейкам памяти, соответствующим «пустым» клеткам, никогда не будет обращения и их можно исключить.

Для построения таблицы без «пустых» клеток определим максимальное и минимальное значения для суммы позиций s:

$$s_{\text{мин}}(\text{k}) = \text{k}\ (\text{k}+1)/2, \qquad s_{\text{макс}} = (\text{k}\ \text{n}) - (\text{k}(\text{k}+1)/2). \tag{9}$$

Перейдем от собственного значения s к его номеру:

$$s1 = s - (k\ (k+1))/2 + 1. \tag{10}$$

Минимальная таблица, после преобразования количества позиций s в номер $s1$ по соотношению (9), приведена на рис. 2c. Нулевая строка соответствующая k=0 и нулевой столбец s=0 исключены, так как данный коэффициент может быть определен на этапе предварительной оценки. Оценим количество ячеек памяти, требуемое для минимальной таблицы. Определим объем V как сумму ненулевых квадратов каждой строки:

$$V = \sum_{k=1}^{n/2} \ ](k\ n - k^2 + 1)/2[.$$

Рассмотрим случай n – четное. Выполнив несложные преобразования, с учетом арифметической прогрессии, получаем соотношение для объема:

$$V = n^3/21 + n^2/16 + n/3 + u/4,$$

где u=0, если n делится на 4, или u=1, если n не делится на 4.

Если n – нечетное, то:

$$V = n^3/24 + 11/24\ n - 1/2.$$

Таким образом, объем минимальной таблицы (рис. 2c) уменьшается в три раза по сравнению с таблицей с учетом симметрии (рис. 2b) или более чем в 12 раз сокращается полная таблица коэффициентов (рис. 2a).

Разрядность q ячеек памяти для записи коэффициентов r в соответствии с (7) определяется следующим соотношением:

$$q =]\ log_2\ (r\ (n,\ ]n/2[,\ ](\ (]n/2[)\ (n+1)\ )/2[)\ +1)[.$$

Проведенный анализ соотношения (8) и проведенные вычисления показали линейную зависимость q от n. В результате для разрядности q получено следующее соотношение:

$$q = n - (2\ log_2\ n - 2).$$

Таким образом, суммарный объем памяти в битах составит V*q.

4. Алгоритм заполнения таблиц

Алгоритм заполнения таблиц основан на последовательном применении для вычисления коэффициентов $r(n, k, s)$ рекуррентного соотношения (2) [22]. В соответствии с преобразованием сумм позиций s в номер $s1$ по соотношению (10), под таблицу отводится объем памяти $V*q$. При этом таблицу можно представить как двумерный массив, который обозначим как A.

Определим значения коэффициентов для простейших случаев. Например, для n=2 A(1,1)=1, A(1,2)=1, A(2,1)=1, остальные значения A принимают нулевые значения. На основании применения соотношения (2) легко определить значения A для n=3 и т.д. Схема перехода для расчета коэффициентов при переходе от размерности (n–1) к размерности n приведена на рис. 3. Вычисления проводятся построчно. Очередная строка из массива A заносится во временный одномерный массив Y, а на ее место заносится строка из временного одномерного массива X. Затем, в соответствии с (2) суммируем элементы $A(k,s1)+Y(s1–n)$ и заносим результат в массив X по адресу $s1$. Далее, если перебрали не все значения $s1$, то увеличиваем $s1$ на 1 и вновь суммируем, если все, то строку k заносим в массив Y, а на ее место заносим строку из массива X и т.д.

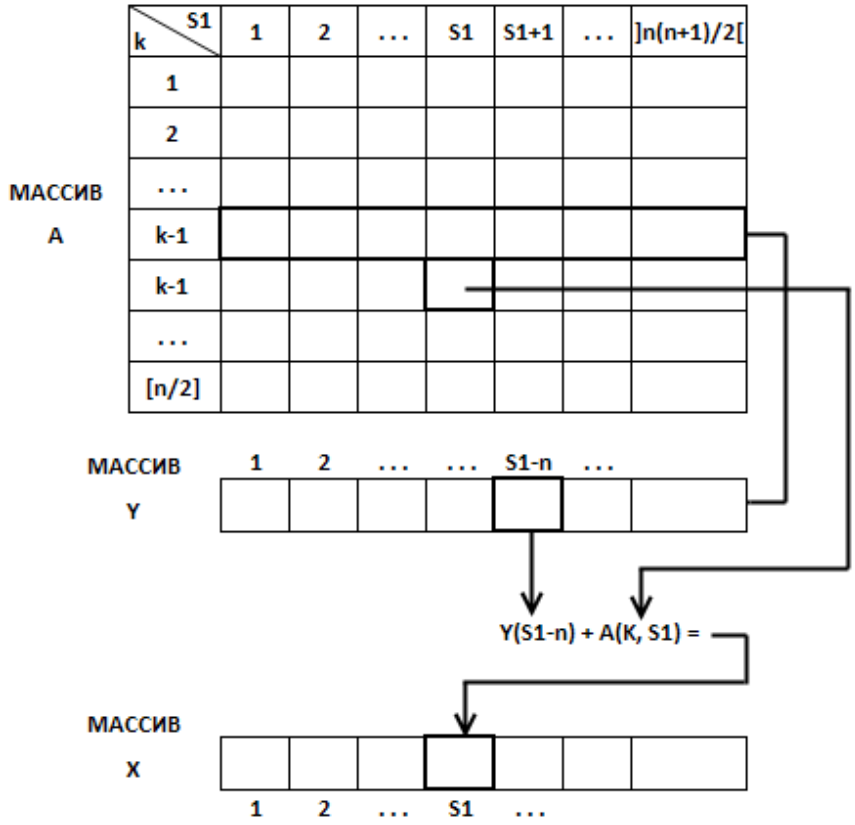

Рис. 3. Схема перехода от размерности (n-1) к размерности n
(Fig. 3. Scheme of transition from dimension (n-1) to dimension n)

На рис. 4 приведен алгоритм заполнения таблицы коэффициентов $r(n, k, s)$. Исходными параметрами являются:

N – текущее значение длины n–блока,

NK – конечное значение длины n–блока,

K1 – максимальное значение k.

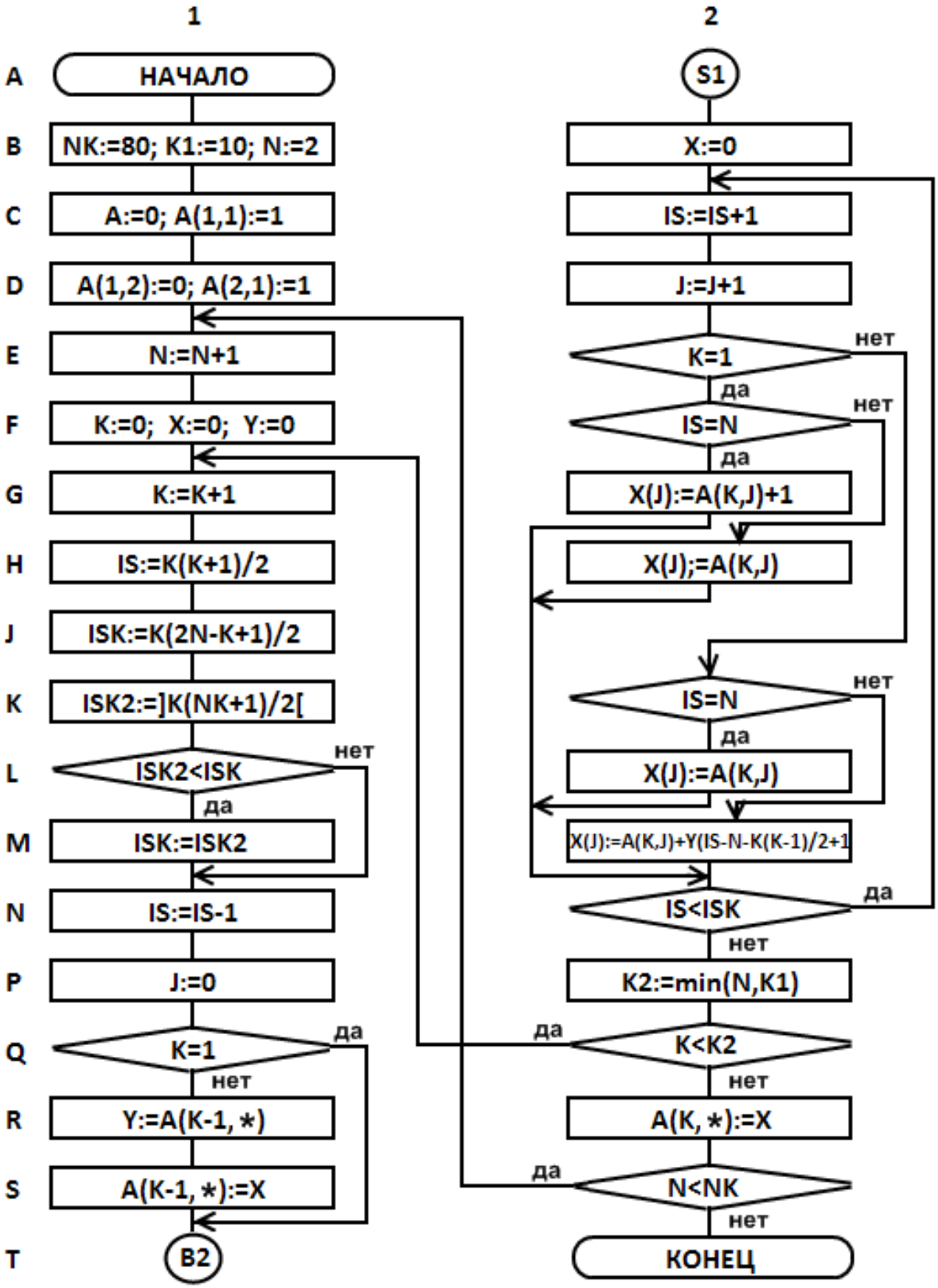

Рис. 4. Алгоритм заполнения таблицы коэффициентов r(n, k, s)
(Fig. 4. Algorithm for filling the coefficient table r(n, k, s)

В алгоритме введены следующие обозначения:
A – двумерный массив коэффициентов,
X и Y – одномерные временные массивы,
K – текущая строка массива A,
IS – текущее минимальное значение *s*,
ISK – текущее максимальное значение *s*,
J – счетная переменная,
K2 –временная переменная.

В начале работы вводятся исходные параметры и задаются начальные значения массивов A=0, X=0, Y=0, начальных коэффициентов A(1,1)=1, A(1,2)=1, A(2,1)=1, переменной K=0.

В блоках алгоритма H1, J1 определяются минимальное IS и максимальное ISK значения сумм позиций *s*. В блоках E2, F2, …, M2 записаны логические условия, при которых второе слагаемое в соотношении (2) принимает значение равное нулю, единице или берется из массива Y.

После завершения работы алгоритма в массиве A будут записаны значения коэффициентов $r(n, k, s)$ соответствующих длине *n*–блока равной NK.

5. Применение кодирования тройками для защиты информации

Универсальное кодирование тройками двоичных наборов (КТ) одновременно с сжатием данных можно использовать для защиты информации.

Алгоритм кодирования опубликован и общеизвестен. Но для того, чтобы правильно декодировать последовательность кодовых слов, ***необходимо знать следующее*** (по сути, совокупность этих сведений является ключом):

1. **Длину исходного блока *n***, т.е. количество двоичных символов в блоках, на которые разбивается исходная двоичная последовательность.

Это основной параметр, без знания которого провести декодирование невозможно. Знание *n* является необходимым, но недостаточным условием для правильной расшифровки, поскольку необходимо знать и другие параметры рассмотренные ниже.

2. **Обычное кодирование или модифицированное.**

В случае обычного кодирования количество разрядов, отводимое в кодовом слове на сумму позиций единиц, постоянно и рассчитано на максимальное возможное значение суммы для данной длины исходного блока – второе слагаемое в формуле (1). При модифицированном кодировании количество разрядов, отводимое на сумму позиций единиц в исходном блоке, становится переменным и зависит от количества единиц в блоке. В этом случае разрядность кодового слова *w* (в битах) будет равна:

$$Z =]log_2 (n+1)[\; + \;]log_2 (k\,(n-k)+1[\; + \;]log_2 r(n, k, s)[.$$

С точки зрения эффективности сжатия данных модифицированное кодирование всегда предпочтительней, но для защиты информации вполне можно использовать и обычное. Без знания, какое кодирование использовано, расшифровка невозможна, даже если известна исходная длина блока.

3. **Применение процедуры адаптации.**

В случае использования адаптации, кодирование применяется для значений *k* меньших или равных k_{max}, а для больших k_{max} *n*–блок передается без кодирования, с соответствующим «флагом» в формате кодового слова *w*. Возможен, также вариант адаптации, когда кодирование применяется и для *k* больших или равных $(n–k_{max})$ с учетом

свойств симметрии кодирования, формулы (5), (6), т.е. сжатие для k такое же, как для (n–k).

4. **Применение инверсии.**

Как отмечено выше кодирование обладает свойством симметрии. Поэтому можно кодировать не двоичный n–блок, а его инверсию. При этом вводится соответствующий «флаг» в формате кодового слова w.

5. **Место расположения «флагов» в кодовом слове.**

Для усложнения кодирования можно размещать «флаги» инверсии и адаптации не в начале кодового слова, а в произвольных разрядах. При этом позиции «флагов» должны быть в пределах минимально возможной длины кодового слова w. Также возможны варианты, когда адаптация и инверсия не применяется или применяется что-то одно.

Следует отметить, что полный перебор в данном случае не поможет, как из-за огромного количества различных вариантов возможных соотношений перечисленных пяти типов параметров ключа, так и из-за того, что просто не с чем сравнивать.

Таким образом, можно утверждать, что кодирование тройками (КТ) не только устраняет статистическую избыточность, но защитит информацию.

Заключение

В ходе данного исследования проведен анализ и ключевые особенности методов сжатия. Описан метод универсального кодирования тройками двоичных наборов для сжатия без потерь в условиях неизвестной статистики источника сообщений. Для формирования коэффициентов кодирования применяется детектирование позиций единиц в n–блоках входной бинарной последовательности. Проведена оценка разрядности кодовых слов при адаптации кодирования к количеству единиц в n–блоках.

Получены соотношения для вычисления коэффициентов кодового слова. Разработан табличный подход для хранения коэффициентов, приводятся варианты формирования таблиц, проведена оценка требуемых объемов памяти для хранения таблиц и показаны подходы для сокращения емкости памяти в 12 раз без потери точности коэффициентов. Разработана методика обработки рекуррентных соотношений для вычисления коэффициентов, обладающая неэкспотенциальной трудоемкостью. Предложен и апробирован алгоритм для заполнения таблиц коэффициентов кодовых слов n–блоков.

Разработаны подходы для защиты данных одновременно с их сжатием без потерь. Для защиты информации предложено пять типов параметров ключей, совокупность которых обладает множеством вариантов при их применении.

СПИСОК ЛИТЕРАТУРЫ:

1. Vavrenyuk A.B., Klarin A.P., Makarov V.V., Shurygin V.A. Data compression methods. Journal of Theoretical and Applied Information Technology. 2015. V. 80 (No 2). P. 423–438. ISSN 1992-8645. E- ISSN 1817-3195. URL: http://www.jatit.org/volumes/Vol80No2/2Vol80No2.pdf. (дата обращения: 05.12.2020).
2. Wayner P. Compression algorithms for real programmers. // Morgan Kaufman, 1999. – 252 p. ISBN: 9780127887746.
3. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии. / Пер. с англ. М. Триумф, 2003. – 319 с. ISBN 5-89392-079-1.
4. Д. Сэломон. Сжатие данных, изображения и звука. М.: Техносфера, 2006. – 368 с. ISBN 5-94836-027-X, 0-387-95260-8.
5. Ковалгин Ю.А., Вологдин Э.И. Цифровое кодирование звуковых сигналов //М.: Корона-Принт, 2004. – 240 с. ISBN: 978-5-7931-0290–2.
6. Salomon D. Data compression: The complete reference. // Springer, 2006. – 1118 p. ISBN: 1-84628-602-6 (978-1-84628-602-5).

7. Соловьев В.Ф. Рациональное кодирование при передаче сообщений. М.: Энергия, Серия: Библиотека по автоматике, вып. 411, 1970. – 64 с.
8. William A. Pearlman, Amir Said. Digital signal compression: Principles and practice // Cambridge University Press, 2011. – 420 p. ISBN: 10-0521805031, 0521899826. URL: https://books.google.ru/books?id=s3H8s8rdsHkC&printsec=frontcover&dq=inauthor:%22William+A.+Pearlman%22&hl=ru&sa=X&ved=2ahUKEwitn7HD3_DtAhUI-yoKHXmfAtoQ6AEwAnoECAEQAg#v=onepage&q&f=false (дата обращения: 05.12.2020).
9. Khalid Sayood. Introduction to data compression (4th Edition)// Printbook, 2012. – 768 p. ISBN: 1849690308.
10. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: Диалог-МИФИ, 2002. – 384 с. ISBN 5-86404-170-X.
11. Новиков Г.Г., Шурыгин В.А., Ядыкин И.М. Устройство для упаковки данных. Патент RU 2701711 C1. Заявка RU 2019100240, 09.01.2019. Опубликовано 30.09.2019. Бюл. №28. МПК H03M 7/30.
12. Новиков Г.Г., Ядыкин И.М. Устройство для распаковки данных. Патент RU 2729509 C1. Заявка RU 2019143298, 23.12.2019. Опубликовано 07.08.2020. Бюл. №28. МПК H03M 7/30.
13. Фитингоф Б.М. Сжатие дискретной информации. Проблемы передачи информации. Т 3. Вып. 3. 1967. С. 28–36. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=1909&wshow=paper (дата обращения: 05.12.2020).
14. Колмогоров А.Н. Три подхода к определению понятия «количество информации». Проблемы передачи информации. Т 1. Вып. 1. 1965. С. 3–11. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=68&wshow=paper (дата обращения: 05.12.2020).
15. Успенский В.А., Вьюгин В.В. Становление алгоритмической теории информации в России. Информационные процессы. Т. 10, № 2, 2010. С. 145–158. URL: http:// www.jip/ru/2010/145-158-2010/pdf (дата обращения: 05.12.2020).
16. Штарьков Ю.М. Универсальное кодирование. Теория и алгоритмы. М.: Физматлит, 2013 – 280 с. ISBN: 9785922115179.
17. Александрович А.Е., Шурыгин В.А., Ядыкин И.М. Метод универсального кодирования двоичных данных. Вопросы радиоэлектроники, серия Электронная вычислительная техника, вып. 2, М.: 2011, С. 94–115.
18. Кларин А.П., Шурыгин В.А. Исследование эффективности универсального кодирования в зависимости от длины блока. / Проблемы передачи информации, 1984. № 2. С. 105–110. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=1136&wshow=paper (дата обращения: 05.12.2020).
19. Виленкин Н.Я., Виленкин А.Н., Виленкин П.А. Комбинаторика. – М.: ФИМА, МЦНМО, 2006. – 400 с. ISBN 5-89492-014-0 («ФИМА»), ISBN5-94057-230-8 (МЦНМО). URL: https://fileskachat.com/download/31544_4f4851eee87775966184b8ca8056bd02.html (дата обращения: 10.12.2020).
20. Справочник по специальным функциям. Под ред. Абрамовица М. и Стиган И. //Пер. с англ. М.: Наука, 1979. – 835 с.
21. Андерсон, Джеймс А. Дискретная математика и комбинаторика. //Пер. с англ. М.: Диалектика / Вильямс, 2020. – 960 с. URL: https://fb2lib.ru/nauchnaya/diskretnaya-matematika-i-kombinatorika/ (дата обращения: 10.12.2020).
22. Швец А.Н. Perl. Примеры программ. URL: http://mech.math.msu.su/~shvetz/54/inf/perl-problems/index.xhtml (дата обращения: 10.12.2020).

REFERENCES:

[1] Vavrenyuk A.B., Klarin A.P., Makarov V.V., Shurygin V.A. Data compression methods. Journal of Theoretical and Applied Information Technology. 2015. V. 80 (no 2). P. 423–438. ISSN 1992-8645. E- ISSN 1817-3195. URL:http://www.jatit.org/volumes/Vol80No2/2Vol80No2.pdf (accessed: 05.12.2020).
[2] Wayner P. Compression algorithms for real programmers. Morgan Kaufman, 1999. – 252 p. ISBN: 9780127887746.
[3] Welstead S. Fractals and wavelets for image compression in action. M.: Triumph, 2003. – 319 p. ISBN 5-89392-079-1 (in Russian).
[4] Salomon D. Compression of data, image and sound. M.: Technosfera, 2006. – 368 p. ISBN 5-94836-027-X, 0-387-95260-8 (in Russian).
[5] Kovalgin Yu.A., Vologdin E.I. Digital coding of audio signal. M.: Korona-Print, 2004. – 240 p. ISBN: 978-5-7931-0290-2 (in Russian).

[6] Salomon D. Data compression: The complete reference. Springer, 2006. – 1118 p. ISBN: 1-84628-602-6 (978-1-84628-602-5.
[7] Solovyev V.F. Efficient coding at transmission of messages. M.: Energy. Series: Library for Automation, issue 411, 1970. – 64 p. (in Russian).
[8] William A. Pearlman, Amir Said. Digital signal compression: Principles and practice. Cambridge University Press, 2011. – 420 p. ISBN: 10- 0521805031, 0521899826. URL:https://books.google.ru/books?id=s3H8s8rdsHkC&printsec=frontcover&dq=inauthor:%22William+A.+Pearlman%22&hl=ru&sa=X&ved=2ahUKEwitn7HD3_DtAhUI-yoKHXmfAtoQ6AEwAnoECAEQAg#v=onepage&q&f=false (accessed: 10.12.2020).
[9] Khalid Sayood. Introduction to data compression (4th Edition). Printbook, 2012. – 768 p. ISBN: 1849690308.
[10] Vatolin D., Ratushniak A., Smirnov M., Yukin V. Data compression methods. Design of archivers, compression of images and video. M.: Dialog-MEPhI, 2002. – 384 p. ISBN 5-86404-170-X (in Russian).
[11] Novikov G.G., Shurygin V.A., Yadykin I.M. Device for packing data. Patent RU 2701711 C1. Application RU 2019100240, 09.01.2019. Published 30.09.2019, bull. no. 28. IPC H03M 7/30 (in Russian).
[12] Novikov G.G., Yadykin I.M. Device for unpacking data. Patent RU 2729509 C1. Application RU 2019143298 23.12.2019. Published 07.08.2020, bull. no. 22. IPC H03M 7/30 (in Russian).
[13] Fitingof B.M. Compression of discrete information. Problemy Peredachi Informatsii. V. 3. no. 3. 1967. P. 28–36. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=1909&wshow=paper. (accessed: 05.12.2020) (in Russian).
[14] Kolmogorov A.N. Three approaches to the definition of «Information amount». Problems of information transmission. Vol. 1, no 1. 1965. P. 3–11. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=68&wshow=paper (accessed: 05.12.2020) (in Russian).
[15] Uspensky V.A., Vyugin V.V. Formation of algorithmic information theory in Russia. Information processes. V. 10, no. 2, 2010. P. 145–158. URL: http:// www.jip/ru/2010/145-158-2010/pdf (дата обращения: 5.12.2020) (accessed: 10.12.2020) (in Russian).
[16] Shtarkov Yu.M. Universal coding. Theory and algorithms. M.: Fizmatlit, 2013. – 280 p. ISBN: 9785922115179 (in Russian).
[17] Aleksandrovich A.E., Shurygin V.A., Yadykin I.M. A method of universal data coding. Questions of radio electronics, a series of Electronic computers. V. 2, 2011. P. 94–115. (in Russian).
[18] Klarin A.P., Shurygin V.A. Analysis of Efficiency of Universal Coding as a Function of the Block Length. Problemy Peredachi Informatsii. V. 20, no. 2. 1984. P. 105–110. URL: http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&option_lang=rus&paperid=1136&wshow=paper (accessed: 05.12.2020) (in Russian).
[19] Vilenkin N.Ya., Vilenkin A.N., Vilenkin P.A. Combinatorics. M.: FIMA, MTsNMO, 2006. ISBN 5-89492-014-0 («FIMA»), ISBN5-94057-230-8 (MTsNMO). URL: https://fileskachat.com/download/31544_4f4851eee87775966184b8ca8056bd02.html (accessed: 10.12.2020) (in Russian).
[20] Abramowitz M., Stegun I.A. Handbook of Mathematical Functions. M.: Nauka, 1979. – 835 p. (in Russian).
[21] Anderson James A. Discrete mathematics and combinatorics. M.: Vil'jams, 2006. – P. 960. URL: https://fb2lib.ru/nauchnaya/diskretnaya-matematika-i-kombinatorika/ (accessed: 10.12.2020) (in Russian).
[22] Shvets A.N. Perl. Sample programs. URL: http://mech.math.msu.su/~shvetz/54/inf/perl-problems/index.xhtml (accessed: 10.12.2020) (in Russian).

Поступила в редакцию – 16 ноября 2020 г. Окончательный вариант – 14 января 2021 г.
Received – November 16, 2020. The final version – January 14, 2021.