УДК 004.624:004.021

Распределение мощности кодов с наименьшей избыточностью алфавитов в зависимости от количества бит и кодового расстояния

А. А. Блюдов, Д. В. Пивоваров, Г. Ю. Пронин

Петербургский государственный университет путей сообщения Императора Александра I, Российская Федерация, 190031, Санкт-Петербург, Московский пр., 9

Для цитирования: *Блюдов А. А., Пивоваров Д. В., Пронин Г. Ю.* Распределение мощности кодов с наименьшей избыточностью алфавитов в зависимости от количества бит и кодового расстояния // Известия Петербургского университета путей сообщения. — СПб.: ПГУПС, 2023. — Т. 20. — Вып. 2. — С. 365-375. DOI: 10.20295/1815-588X-2023-2-365-375

Аннотация

Цель: Исследовать зависимость максимальной мощности кодов от количества разрядов и минимального кодового расстояния; найти подход к определению оптимальных правил построения контрольного вектора разделимого кода с точки зрения обеспечения минимальной избыточности при заданной достоверности передачи сообщения. **Методы:** Для проведения экспериментальных исследований использовалось компьютерное моделирование. Для теоретических исследований применены метод аналитического обзора, теория графов, теория кодирования. **Результаты:** Теоретически и экспериментально получены некоторые частные случаи распределения максимальной мощности кодовых алфавитов с заданным расстоянием Хэмминга для различных постоянных длин, полученных с помощью ранее описанного алгоритма. Предложен и описан метод удвоения мощности произвольных двоичных кодов, а также способ получения кодов с наименьшей избыточностью мощностей $M = 2^f$, где f — натуральное число, для заранее заданного минимального кодового расстояния путем рекурсивного использования предложенного в статье метода. **Практическая значимость:** Выработан алгоритм удвоения мощности кодового алфавита при сохранении требуемой достоверности передачи данных. Получена методика анализа получаемых матриц кодовых векторов с целью определения правил вычисления контрольных разрядов без использования циклических алгоритмов.

Ключевые слова: Помехозащитное кодирование, коды с наименьшей избыточностью, кодовый алфавит, кодовое слово, расстояние Хэмминга, разделимые коды.

Введение

При передаче информации по двоичному каналу с симметричным распределением ошибок зачастую возникает необходимость в применении помехозащитного кодирования. В статье рассматриваются только те случаи, при которых в передаваемых сообщениях присутствует постоянное число бит. При решении задачи выбора кода, наилучшим образом удовлетворяющего задачам обнаружения и исправления ошибок, исходят из многих параметров. В их числе: количество информации, передаваемое за одно сообщение, требуемая достоверность при приеме сообщения, свойства помех в канале. Так как это далеко не весь список параметров, на основании которых делают выбор кода, а также рассматриваемая авторами статьи модель (двоичный симметричный канал, равномерное блочное кодирование) применяется не во всех случаях, воспользуемся некоторым упрощением. В качестве канала и способа кодирования будем использовать только вышеупомянутые, а требуемая достоверность

передачи и свойства помехи будут определять требуемое минимальное кодовое расстояние для любой длины кодового слова.

Теперь постановка задачи выбора кода выглядит следующим образом: требуется передать M сообщений, закодированных кодом с определенным расстоянием Хэмминга. Код, удовлетворяющий этим условиям, обеспечит возможность передать любую команду из M, а соблюдение минимального кодового расстояния $d_{\min}$ между всеми словами исключает возможность «перепутать» команды вследствие ошибки передачи [1]. При этом естественным образом возникает задача определения способа кодирования таким образом, чтобы расстояние Хэмминга было не меньше заданного, но избыточность кода была минимальной.

В данной статье приводятся некоторые результаты ранее проведенных исследований, полученные экспериментально. На их основании сделан ряд наблюдений, подтвержденных расчетно, на основании которых, в свою очередь, был предложен и описан метод удвоения мощности кодовых алфавитов, с помощью которого можно получить коды для использования при передаче данных в системах автоматики и телемеханики. Эти коды при равных расстояниях Хэмминга не будут уступать по мощности известным классам двоичных кодов. Преимущество предлагаемых кодов по сравнению с известными циклическими кодами состоит в простоте процедур кодирования и декодирования аппаратным методом и программируемых логических интегральных схем.

1. Зависимость мощности кодовых алфавитов от количества разрядов и минимального кодового расстояния

При написании этой статьи авторы исходили из следующих допущений:

1. Передача информации происходит по двоичному симметричному каналу.

2. При передаче используется блочное равномерное кодирование.

3. При выборе кода определяющими параметрами являются требуемая мощность кодового алфавита и вероятность успешной передачи информации в пакете.

Количество разрядов в кодовом слове обозначим как n, расстояние Хэмминга между двумя кодовыми словами — d. Каждому значению n соответствует полносвязный граф, у которого 2^n вершин, а вес ребер соответствует расстоянию Хэмминга. Задача поиска кода наибольшей мощности для заданных n и d соответствует задаче поиска остова заданного веса. Так как графы, являющиеся представлением всех двоичных кодовых слов длины n, являются частными случаями, то для них можно использовать алгоритм, который не учитывает такие обстоятельства, как: циклы с отрицательной суммой весов, петли с нулевыми расстояниями и т. д.

Обозначим кодовое расстояние между векторами a_i и b_j одинаковой длины как $d(a_i, b_j)$, где i и j — номера элементов в множествах I и J соответственно.

Алгоритм [2]. *Нахождение подграфа максимальной мощности с заданным кодовым расстоянием.*

1. Множество I устанавливаем пустым.

2. Множество J возможных кодовых слов длины n пронумеруем в соответствии с десятичным значением кодовых слов.

3. Элемент с нулевым номером переносим в множество I.

4. Выбираем следующий элемент b с номером $j + 1$.

5. Если выполняется условие $d(a_i, b_j) \geq c$, где c — заранее определенное минимальное кодовое расстояние, для любого i, то переносим элемент b в кодовое множество I;

6. После того как все элементы множества J проверены, множество I — искомый код

ТАБЛИЦА 1. Зависимость мощностей кодовых алфавитов M в зависимости от n и d_{min}

Максимальная мощность кодовых алфавитов												
n \ d	1	2	3	4	5	6	7	8	9	10	11	12
1	2											
2	4	2										
3	8	4	2									
4	16	8	2	2								
5	32	16	4	2	2							
6	64	32	8	4	2	2						
7	128	64	16	8	2	2	2					
8	256	128	16	16	4	2	2	2				
9	512	256	32	16	4	4	2	2	2			
10	1024	512	64	32	8	4	2	2	2	2		
11	2048	1024	128	64	16	8	4	2	2	2	2	
12	4096	2048	256	128	16	16	4	4	2	2	2	2
13	8192	4096	512	256	32	16	8	4	2	2	2	2
14	16 384	8192	1024	512	64	32	16	8	4	2	2	2
15	32 768	16 384	2048	1024	128	64	32	16	4	4	2	2
16	65 536	32 768	2048	2048	256	128	32	32	4	4	2	2
17	131 072	65 536	4096	2048	512	256	64	32	8	4	4	2

Результаты расчетов мощностей согласно алгоритму представлены в табл. 1.

Из полученных данных видно, что все мощности M, указанные в таблице, являются степенями числа 2 вне зависимости от числа разрядов в блоке и кодового расстояния.

Рассмотрим пример поиска алфавитов максимальной мощности с $d_{min} = 3$ для $n = 3$ и $n = 5$. Выберем в качестве первого элемента слово <000>. В дальнейшем для упрощения записи кодовых слов будем использовать десятичные эквиваленты кодовых слов в фигурных скобках; например {0} вместо <000>. Далее для каждого кодового слова сравним расстояние Хэмминга с заданным d_{min}. Если выполняется условие $d(0, j) \geq d_{min}$, где j — кодовое слово, то добавим его в алфавит. Так как элемент {7} содержит наибольшее количество единиц, то не существует других кодовых слов, удовлетворяющих требованию минимального кодового расстояния. Таким образом, в результате действия алгоритма получен кодовый алфавит с параметрами $n = 3$, $M = 2$, $d_{min} = 3$, {0,7} в десятичной записи. Матрица расстояний Хэмминга для $n = 3$ представлена в табл. 2.

ТАБЛИЦА 2. Матрица расстояний Хэмминга для $n = 3$

	0	1	2	3	4	5	6	**7**
0	**0**	1	1	2	1	2	2	**3**
1	1	0	2	1	2	1	3	2
2	1	2	0	1	2	3	1	2
3	2	1	1	0	3	2	2	1
4	1	2	2	3	0	1	1	2
5	2	1	3	2	1	0	2	1
6	2	3	1	2	1	2	0	1
7	3	2	2	1	2	1	1	**0**

Теперь рассмотрим работу алгоритма для $n = 5$. Матрица расстояний Хэмминга для $n = 5$ представлена в табл. 3. В качестве первого элемента выберем {0}. Далее работа аналогична до момента, когда известны два слова из кодового алфавита {0, 7}. Далее условием включения слова в искомое подмножество является $d(0, j) \geq 3 \wedge d(7, j) \geq 3$. Этому условию соответствует {25}. Добавляем его в кодовый алфавит {0, 7, 25}. Условием для включения следующего элемента будет $d(0, j) \geq 3 \wedge$

$\wedge\, d(7, j) \geq 3 \wedge d(25, j) \geq 3$. Ему соответствует {30}. Теперь кодовый алфавит выглядит следующим образом {0, 7, 25, 30}. Ему соответствует новое условие $d(0, j) \geq 3 \wedge d(7, j) \geq 3 \wedge (25, j) \geq 3 \wedge d(30, j) \geq 3$. Единственный оставшийся элемент {31} ему не соответствует.

В полученном алфавите мощностью $M = 4$ есть кодовые слова, представление которых в десятичной форме совпадает с аналогичным представлением алфавита $M = 2$. Их двоичные представления для $M = 4$ и $M = 2$ соответственно: {00000, 00111, 11010, 11101} и {000, 111}. Добавление нулей (равно как и единиц) в одноименные позиции слов алфавита с меньшим n позволяет получить обладающие аналогичными свойствами слова алфавита с большим n. При этом появляется возможность увеличить мощность полученного алфавита, добавив в него новые слова при сохранении $d_{\min}$.

2. Алгоритм удвоения мощности кодового алфавита

Для определения зависимости M от n и $d_{\min}$ рассмотрим некоторые частные случаи.

Так, алфавиты первого столбца $d = 1$ представляют собой все возможные кодовые слова заданной длины, и их мощности определяются как $M = 2^n$.

Второй столбец $d = 2$ соответствует способу кодирования «бит четности/нечетности». Для него соотношение M и n выглядит следующим образом:

$$M = 2^{n-1}. \tag{1}$$

Третий столбец соответствует классическому коду Хэмминга [3–6].

Однако не для каждого значения d существует код с уже определенными правилами построения, поэтому рассмотрим иной способ определения зависимости. Для кодов с $M = 2$, у которых минимальное из возможных n (верхние в табл. 1), верно соотношение $n = d$. Эти алфавиты представляют собой произвольное слово длины d и его инверсию [4].

Так как каждый из алфавитов большей мощности в два раза превосходит алфавит меньшей мощности с сохранением минимального кодового расстояния, то существует как минимум один алгоритм увеличения мощности. Для описания предложенного алгоритма введем ряд обозначений и операций.

Определим операцию $A \oplus c = B$, где A — кодовый алфавит, a_i — вектор-строка, являющаяся i-м элементом алфавита A, c — вектор-строка того же размера, что и a_i. Результатом операции будет кодовый алфавит B вида:

$$B = \begin{vmatrix} a_1 \oplus c \\ \vdots \\ a_n \oplus c \end{vmatrix}. \tag{2}$$

Так как операция сложения по модулю два только инвертирует некоторые одноименные биты во всех словах a_i, такое действие не изменит свойства кода по достоверности передачи информации.

Под операцией $H(A, B)$ будем понимать матрицу кодовых расстояний:

$$H(A, B) = \begin{pmatrix} d(a_1, b_1) & \cdots & d(a_1, b_n) \\ \vdots & \ddots & \vdots \\ d(a_n, b_1) & \cdots & d(a_n, b_n) \end{pmatrix}, \tag{3}$$

где A и B — равномощные кодовые алфавиты, а их элементы a_i и b_i, соответственно, вектор-строки равной длины.

Определим операцию ADD(A, B) = C, где A и B — кодовые алфавиты с одинаковым значением n, a_i и b_j — их элементы, а кодовый алфавит C:

$$C = \begin{vmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{vmatrix}. \tag{4}$$

ТАБЛИЦА 3. Матрица расстояний Хэмминга для $n = 5$

	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	29	**30**	31
0	0	1	1	2	1	2	2	**3**	1	2	2	3	2	3	3	4	1	2	2	3	2	3	3	4	2	**3**	3	4	3	4	**4**	5
1	1	0	2	1	2	1	3	2	2	1	3	2	3	2	4	3	2	1	3	2	3	2	4	3	3	2	4	3	4	3	5	4
2	1	2	0	1	2	3	1	2	2	3	1	2	3	4	2	3	2	3	1	2	3	4	2	3	3	4	2	3	4	5	3	4
3	2	1	1	0	3	2	2	1	3	2	2	1	4	3	3	2	3	2	2	1	4	3	3	2	4	3	3	2	5	4	4	3
4	1	2	2	3	0	1	1	2	2	3	3	4	1	2	2	3	2	3	3	4	1	2	2	3	3	4	4	5	2	3	3	4
5	2	1	3	2	1	0	2	1	3	2	4	3	2	1	3	2	3	2	4	3	2	1	3	2	4	3	5	4	3	2	4	3
6	2	3	1	2	1	2	0	1	3	4	2	3	2	3	1	2	3	4	2	3	2	3	1	2	4	5	3	4	3	4	2	3
7	3	2	2	1	2	1	1	**0**	4	3	3	2	3	2	2	1	4	3	3	2	3	2	2	1	5	**4**	4	3	4	3	**3**	2
8	1	2	2	3	2	3	3	4	0	1	1	2	1	2	2	3	2	3	3	4	3	4	4	5	1	2	2	3	2	3	3	4
9	2	1	3	2	3	2	4	3	1	0	2	1	2	1	3	2	3	2	4	3	4	3	5	4	2	1	3	2	3	2	4	3
10	2	3	1	2	3	4	2	3	1	2	0	1	2	3	1	2	3	4	2	3	4	5	3	4	2	3	1	2	3	4	2	3
11	3	2	2	1	4	3	3	2	2	1	1	0	3	2	2	1	4	3	3	2	5	4	4	3	3	2	2	1	4	3	3	2
12	2	3	3	4	1	2	2	3	1	2	2	3	0	1	1	2	3	4	4	5	2	3	3	4	2	3	3	4	1	2	2	3
13	3	2	4	3	2	1	3	2	2	1	3	2	1	0	2	1	4	3	5	4	3	2	4	3	3	2	4	3	2	1	3	2
14	3	4	2	3	2	3	1	2	2	3	1	2	1	2	0	1	4	5	3	4	3	4	2	3	3	4	2	3	2	3	1	2
15	4	3	3	2	3	2	2	1	3	2	2	1	2	1	1	0	5	4	4	3	4	3	3	2	4	3	3	2	3	2	2	1
16	1	2	2	3	2	3	3	4	2	3	3	4	3	4	4	5	0	1	1	2	1	2	2	3	1	2	2	3	2	3	3	4
17	2	1	3	2	3	2	4	3	3	2	4	3	4	3	5	4	1	0	2	1	2	1	3	2	2	1	3	2	3	2	4	3
18	2	3	1	2	3	4	2	3	3	4	2	3	4	5	3	4	1	2	0	1	2	3	1	2	2	3	1	2	3	4	2	3
19	3	2	2	1	4	3	3	2	4	3	3	2	5	4	4	3	2	1	1	0	3	2	2	1	3	2	2	1	4	3	3	2
20	2	3	3	4	1	2	2	3	3	4	4	5	2	3	3	4	1	2	2	3	0	1	1	2	2	3	3	4	1	2	2	3
21	3	2	4	3	2	1	3	2	4	3	5	4	3	2	4	3	2	1	3	2	1	0	2	1	3	2	4	3	2	1	3	2
22	3	4	2	3	2	3	1	2	4	5	3	4	3	4	2	3	2	3	1	2	1	2	0	1	3	4	2	3	2	3	1	2
23	4	3	3	2	3	2	2	1	5	4	4	3	4	3	3	2	3	2	2	1	2	1	1	0	4	3	3	2	3	2	2	1
24	2	3	3	4	3	4	4	5	1	2	2	3	2	3	3	4	1	2	2	3	2	3	3	4	0	1	1	2	1	2	2	3
25	3	2	4	3	4	3	5	4	2	1	3	2	3	2	4	3	2	1	3	2	3	2	4	3	1	**0**	2	1	2	1	**3**	2
26	3	4	2	3	4	5	3	4	2	3	1	2	3	4	2	3	2	3	1	2	3	4	2	3	1	2	0	1	2	3	1	2
27	4	3	3	2	5	4	4	3	3	2	2	1	4	3	3	2	3	2	2	1	4	3	3	2	2	1	1	0	3	2	2	1
28	3	4	4	5	2	3	3	4	2	3	3	4	1	2	2	3	2	3	3	4	1	2	2	3	1	2	2	3	0	1	1	2
29	4	3	5	4	3	2	4	3	3	2	4	3	2	1	3	2	3	2	4	3	2	1	3	2	2	1	3	2	1	0	2	1
30	4	5	3	4	3	4	2	3	3	4	2	3	2	3	1	2	3	4	2	3	2	3	1	2	2	3	1	2	1	2	**0**	1
31	5	4	4	3	4	3	3	2	4	3	3	2	3	2	2	1	4	3	3	2	3	2	2	1	3	2	2	1	2	1	1	0

Введем действие $UP(A, B) = D$, где A и B — кодовые алфавиты равной мощности M, $a_i(x_1, \ldots, x_k)$ и $b_i(y_1, \ldots, y_n)$ соответственно их элементы с количеством разрядов k и n.

$$D = \begin{vmatrix} (x_1 \cdots x_k y_1 \cdots y_n)_1 \\ \vdots \\ (x_1 \cdots x_k y_1 \cdots y_n)_M \end{vmatrix}. \qquad (5)$$

Обозначим матрицу I^e_M, где M — количество строк матрицы, e — число столбцов, x_{Me} — элемент матрицы.

$$I^e_M = \begin{vmatrix} x_{11} & \cdots & x_{1e} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{Me} \end{vmatrix} x_{Me} = \begin{cases} 0 < \dfrac{M}{2} \\ 1\,\dfrac{M}{2} \end{cases}. \qquad (6)$$

Далее рассмотрим предпосылки, из которых будем исходить при создании алгоритма. В качестве отправной точки имеем кодовый алфавит A с максимально возможным $M = 2^f$, где f — натуральное число, n и $d_{\min}$. Получим алфавит $B = A \oplus c$. Для этого подберем такое c, чтобы получить набольший из возможных $\min(H(A, B))$. Далее составим из них новый алфавит $C = ADD(A, B)$. Кодовые расстояния между элементами алфавита C:

$$H(C,C) = \begin{pmatrix} H(A,A) & H(A,B) \\ H(B,A) & H(B,B) \end{pmatrix}. \tag{7}$$

Так как $d(a_i, b_i)$ коммутативно, то $H(A, B) = = H(B, A)$ [1]. Минимальное кодовое расстояние в алфавите A равно $d_{\min}$ по определению. В алфавите B оно также равно $d_{\min}$, поскольку сложение каждого элемента с произвольной константой не изменяет минимальное кодовое расстояние в алфавите. Значит, минимальное кодовое расстояние алфавита C — это $\min(H(A, B))$. $\min(H(A, B)) < d_{\min}$, так как если бы он мог быть ему равен, то мощность алфавита C была бы больше, чем у A, а это невозможно по определению A. Матрица кодовых расстояний I^e_{2M} выглядит следующим образом:

$$H(I^e_{2M}, I^e_{2M}) = \begin{pmatrix} 0_{11} & \cdots & 0_{1(M/2)} & e_{1(M/2+1)} & \cdots & e_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0_{(M/2)1} & \cdots & 0_{(M/2)(M/2)} & e_{(M/2)(M/2+1)} & \cdots & e_{(M/2)M} \\ e_{(M/2+1)1} & \cdots & e_{(M/2+1)(M/2)} & 0_{(M/2+1)(M/2+1)} & \cdots & 0_{(M/2+1)M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{M1} & \cdots & e_{M(M/2)} & 0_{M(M/2+1)} & \cdots & 0_{MM} \end{pmatrix}. \tag{8}$$

Места с ненулевыми значениями в этой матрице соответствуют позициям $H(A, B)$ $H(B, A)$. Необходимое значение $e = d_{\min} - \min(H(A,B))$. Учитывая это, получим алфавит D по формуле:

$$D = UP(I^e_{2M}, ADD(A, B)). \tag{9}$$

В общем виде формула для увеличения алфавита удвоенной мощности выглядит так:

$$D = UP(I^e_{2M}, ADD(A, A \oplus c)). \tag{10}$$

3. Частные случаи зависимости максимальной мощности кодовых алфавитов от общего числа бит и минимального кодового расстояния

Все нижеописанные зависимости верны для кодов с минимальной избыточностью с наименьшим числом разрядов при одинаковом минимальном кодовом расстоянии и равных мощностях. Для большей наглядности приведем примеры расширения мощностей алфавитов с $d_{\min} = 3$. В качестве начального алфавита используем A_2:

$$A_2 = \begin{vmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}. \tag{11}$$

Теперь получим алфавит со свойствами $M = 4$, $d_{\min} = 3$. Подходящие константы в десятичной записи это $\{1, 2, 3, 4, 5, 6\}$. Для простоты выберем $c = 001$. Тогда B_2:

$$B_2 = A_2 \oplus (0,0,1) = \\ = \begin{vmatrix} 0\oplus 0 & 0\oplus 0 & 0\oplus 1 \\ 1\oplus 0 & 1\oplus 0 & 1\oplus 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}. \quad (12)$$

Тогда $H(A_2, B_2)$:

$$H\left(A_2, B_2\right) = \begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix}. \quad (13)$$

Отсюда следует, что:

$$e = d_{\min} - \min\left(H\left(A_2, B_2\right)\right) = 3 - 1 = 2. \quad (14)$$

Значит, искомый алфавит A_4 можно получить по формуле:

$$A_4 = UP\left(I_4^2, ADD\left(A_2, B_2\right)\right), \quad (15)$$

где разряды, которые необходимо добавить, определяются как I_4^2:

$$I_4^2 = \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{vmatrix}. \quad (16)$$

А результат совмещения кодовых алфавитов $APP(A_2, B_2)$:

$$ADD\left(A_2, B_2\right) = \begin{vmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}. \quad (17)$$

Полученный таким образом код A_4 представляет собой:

$$A_4 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix}. \quad (18)$$

Теперь рассмотрим более общий случай удвоения алфавитов мощностью $M = 2$ при произвольном $d_{\min}$. Обозначим такой алфавит как E_2. Количество разрядов для него $n = d_{\min}$. Существует 2^n возможных вариантов алфавита, однако все они могут быть сведены путем операций, не изменяющих кодовое расстояние и число бит, таких как перестановка строк, столбцов, инверсия разрядов, к алфавиту из двух кодовых слов, представляющих собой n нулей и n единиц. Матрица кодовых расстояний такого алфавита:

$$H\left(E_2, E_2\right) = \begin{vmatrix} 0 & d \\ d & 0 \end{vmatrix}. \quad (19)$$

Далее необходимо выбрать такую константу c, чтобы максимизировать $\min H(E2, E2 \oplus c)$. Данному условию будут удовлетворять все константы, в которых число единиц равно половине кодового расстояния. Тогда:

$$H\left(E_2, E_2 \oplus c\right) = \begin{Vmatrix} \left\lfloor \frac{d}{2} \right\rfloor & \left\lceil \frac{d}{2} \right\rceil \\ \left\lceil \frac{d}{2} \right\rceil & \left\lfloor \frac{d}{2} \right\rfloor \end{Vmatrix}. \quad (20)$$

Под $\lfloor\ \rfloor$ и $\lceil\ \rceil$ тут понимается округление до целого числа вниз и вверх соответственно. Число разрядов, которое необходимо добавить, чтобы сохранить минимальное кодовое расстояние $e = d_{\min} - \operatorname{lfloor}\frac{d}{2}\operatorname{rfloor}$. Следовательно, минимальное n для кодов с наименьшей избыточностью с $M = 4$:

$$n = d_{\min} + e = 2d_{\min} - \left\lfloor \frac{d_{\min}}{2} \right\rfloor = d_{\min} + \left\lceil \frac{d_{\min}}{2} \right\rceil. \quad (21)$$

Продолжая предыдущие операции, получим алфавит E_8 путем удвоения мощности алфавита E_4. Для этого определим вид подходящей константы:

$$E_4 = \begin{vmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & 0 & \cdots & 0 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \end{vmatrix}. \quad (22)$$

Для кодовых расстояний больше 2 в E_4 существует три вида разрядов. Для того чтобы максимизировать $\min(H(E4, E4 \oplus c))$, нужно инвертировать половину разрядов каждого вида. Первая группа представляет собой разряд, который был добавлен для сохранения кодового расстояния. Их количество — e. Вторая группа — это те разряды, которые не инвертировались в результате $E_2 \oplus c$. Это значит, что их $\frac{d_{\min}}{2}$. Третья группа — это разряды, которые находятся на позициях, где происходила инверсия в результате $E_2 \oplus c$. Они присутствуют в алфавите E_2 в количестве $\frac{d_{\min}}{2}$. Значит, количество разрядов, которое необходимо добавить для сохранения $d_{\min}$:

$$e = d_{\min} - 2\left\lfloor \frac{\left\lceil \frac{d_{\min}}{2} \right\rceil}{2} \right\rfloor - \left\lfloor \frac{\left\lfloor \frac{d_{\min}}{2} \right\rfloor}{2} \right\rfloor. \quad (23)$$

Отсюда следует, что минимальное необходимое число разрядов для кодов с $M = 8$:

$$\begin{aligned} n &= d + \left(d - \left\lfloor \frac{d}{2} \right\rfloor \right) + \\ &+ \left(d - \left\lfloor \frac{\left\lceil \frac{d}{2} \right\rceil}{2} \right\rfloor - \left\lfloor \frac{\left\lceil \frac{d}{2} \right\rceil}{2} \right\rfloor - \left\lfloor \frac{\left\lfloor \frac{d}{2} \right\rfloor}{2} \right\rfloor \right) = \\ &= 2d + \left\lceil \frac{d}{2} \right\rceil - 2\left\lfloor \frac{\left\lceil \frac{d}{2} \right\rceil}{2} \right\rfloor - \left\lfloor \frac{\left\lfloor \frac{d}{2} \right\rfloor}{2} \right\rfloor. \end{aligned} \quad (24)$$

Приведем пример удвоения мощности алфавита со свойствами $M = 4$, $d_{\min} = 3$, $n = 5$. Примем $c = \{01010\}$. Тогда:

$$B_4 = \begin{vmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{vmatrix}. \quad (25)$$

$$H\left(A_4, B_4\right) = \begin{pmatrix} 2 & 3 & 3 & 2 \\ 3 & 2 & 2 & 3 \\ 3 & 2 & 2 & 3 \\ 2 & 3 & 3 & 3 \end{pmatrix}. \quad (26)$$

$$e = 3 - 2 = 1. \quad (27)$$

$$A_8 = UP\left(I_8^1, ADD\left(A_4, B_4\right)\right). \quad (28)$$

$$A_8 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{vmatrix}. \quad (29)$$

4. Метод определения информационных и контрольных разрядов в кодах с наименьшей избыточностью

Выше описан метод определения максимальной мощности множества кодовых комбинаций с заданными длиной и кодовым расстоянием, а также итеративного удвоения указанной мощности. Однако решение такой задачи является только средством достижения более важного и прикладного результата — синтеза правил построения разделимых кодов, обладающих соответствующими свойствами. Под синтезом понимается

выделение в получившемся кодовом алфавите A_i информационных и контрольных разрядов.

Опишем разряды в вышеописанных кодах. Для этого определим информационные разряды как:

$$i_j^k = \begin{vmatrix} x_1 \\ \vdots \\ x_j \end{vmatrix}, \qquad (30)$$

где $\tilde{o}_j = \left\lfloor \frac{j-1}{k} \right\rfloor \bmod k.$

На начальном этапе составления кода имеется несколько разрядов вида I_2^1. В тех позициях, где значение константы равно «0», в новом коде также будут разряды вида i_4^1:

$$i_4^1 = \begin{vmatrix} 0 \\ 1 \\ 0 \\ 1 \end{vmatrix}. \qquad (31)$$

Добавленные разряды в результате $UP(I_4, C)$ будут разрядами вида i^2. Те же разряды, которые инвертировались константой, возможно вычислить по формуле $k^{12} = i^1 \oplus i^2$.

Определение разрядов при дальнейших увеличениях будет аналогично: если разряд не инвертировался константой, то он определяется так же, как и ранее, добавленные разряды будут представлять из себя информационные разряды вида i^{k+1}, где k — номер старшего информационного разряда. В случае инверсии разряда его значение определяется как ранее известное, сложенное по модулю два с i^{k+1}. Для наглядности приведем пример в табл. 4.

Многие известные классы двоичных блочных кодов будут совпадать с кодами, которые можно получить описанным в статье методом, по мощности и минимальному кодовому расстоянию. Некоторые из этих кодов являются циклическими, а значит, для их кодирования/декодирования применяются циклические алгоритмы. Для описанных в статье кодов контрольные разряды определяются как сложение по модулю 2 нескольких информационных. При использовании аппаратных методов кодирования/декодирования или в случае использования ПЛИС (программируемых логических интегральных схем) эти операции можно осуществить за один такт, что может дать выигрыш в скорости.

ТАБЛИЦА 4. Пример определения принадлежности разрядов

Номера разряда	6	5	4	3	2	1
Разряды				i^1	i^1	i^1
Константа				0	0	1
Разряды		i^2	i^2	i^1	i^1	$i^1 \oplus i^2$
Константа		0	1	0	1	0
Разряды	i^3	i^2	$i^2 \oplus i^3$	i^1	$i^1 \oplus i^3$	$i^1 \oplus i^2$

Заключение

Метод удвоения мощности кодовых алфавитов, описанный в статье, применим к любому двоичному блочному коду. Новый код, полученный таким образом, гарантированно сохранит минимальное кодовое расстояние, однако при использовании произвольного кодового алфавита в результате удвоения необязательно получится код с наименьшей избыточностью.

Предложенный в статье метод позволяет получить коды с наименьшей избыточностью с произвольным кодовым расстоянием и мощностью $M = 2^f$, где f — натуральное число, путем рекурсивно использования метода, на кодовом алфавите, представляющем собой произвольное слово длинной $d_{\min}$ и его инверсию.

Путем анализа полученных таким образом кодов с наименьшей избыточностью могут быть выявлены правила построения их разрядов, что даст возможность их использования в системах связи и функционального контроля, в том числе и в устройствах автоматики и телемеханики на железнодорожном транспорте [7–10].

На основании исследования метода удвоения мощности были получены аналитические зависимости минимального необходимого числа разрядов от минимального кодового расстояния для алфавитов с мощностями 4 и 8.

Описанный в статье подход позволяет проводить дальнейшие исследования для алфавитов других мощностей.

Библиографический список

1. Hamming R. W. Error detecting and error correcting codes / R. W. Hamming // The Bell system technical journal. — 1950. — Vol. 29. — Iss. 2. — DOI: 10.1002/j.1538-7305.1950.tb00463.x.

2. Пронин Г. Ю. Алгоритм генерации условно оптимальных кодов / Г. Ю. Пронин, Е. А. Волков, Ю. В. Иванов и др. // Транспорт: проблемы, идеи, перспективы. — 2022. — № 2. — С. 96–98.

3. Manoj S. G. Diagonal Hamming Based Multi-Bit Error Detection and Correction Technique for Memories / S. G. Manoj, A. K. Mohan, N. L. Sri Ganesh et al. // International Conference on Communication and Signal Processing. — July 28–30, 2020. — DOI: 10.1109/ICCSP48568.2020.9182249.

4. Van W. J. Effectiveness of Hamming Single Error Correction Codes Under Harsh Electromagnetic Disturbances / W. J. Van, J. Lannoo, J. Vankeirsbilck et al. // International Symposium on Electromagnetic Compatibility. — August 27–30, 2018. — DOI: 10.1109/EMCEurope.2018.8485176.

5. Tshagharyan G. Experimental Study on Hamming and Hsiao Codes in the Context of Embedded Applications / G. Tshagharyan, G. Hatutyunyan, S. Shoukourian et al. // Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, September 29 — October 2, 2017.

6. Musayelyan R. Hamming Distance Based Data Correction Combined With Low Power XOR Circuit / R. Musayelyan // Proceedings of 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021.

7. Sridevi N. Implementation of Error Correction Techniques in Memory Applications / N. Sridevi, K. Jamal, K. Mannem // 5th International Conference on Computing Methodologies and Communication. — April 8–10, 2021. — DOI: 10.1109/ICCMC51019.2021.9418432.

8. Tolentino L. K. S. Overhead Interspersing of Redundancy Bits Reduction Algorithm by Enhanced Error Detection Correction Code / L. K. S. Tolentino, I. C. Valenzuela, R. O. Serfa Juan // Journal of Engineering Science & Technology Review. — 2019. — T. 12. — Iss. 2.

9. Shahariar Parvez A. H. M. Design and implementation of hamming encoder and decoder over FPGA / A. H. M. Shahariar Parvez et al. // International Conference on Computer Networks and Communication Technologies: ICCNCT 2018. — Springer Singapore, 2019. — Pp. 1005–1022.

10. Panem C. Polynomials in Error Detection and Correction in Data Communication System / C. Panem, V. Gad, R. Gad // Coding Theory. — 2019. — P. 29.

Дата поступления: 14.04.2023
Решение о публикации: 20.05.2023

Контактная информация:
БЛЮДОВ Антон Александрович — канд. техн. наук, доц.; blyudov@pgups.ru
ПИВОВАРОВ Дмитрий Вячеславович — канд. техн. наук, доц.; pivovarov.d.v.spb@gmail.com
ПРОНИН Георгий Юрьевич — аспирант; georgiy3pronin@gmail.com

Power Distribution of Codes with the Lowest Alphabet Redundancy Depending on the Number of Bits and Code Distance

A. A. Blyudov, D. V. Pivovarov, G. Yu. Pronin

Emperor Alexander I St. Petersburg State Transport University, 9, Moskovsky pr., Saint Petersburg, 190031, Russian Federation

For citation: Blyudov A. A., Pivovarov D. V., Pronin G. Yu. Power Distribution of Codes with the Lowest Alphabet Redundancy Depending on the Number of Bits and Code Distance // *Proceedings of Petersburg Transport University*, 2023, vol. 20, iss. 2, pp. 365–375. (In Russian). DOI: 10.20295/1815-588X-2023-2-365-375

Summary
Purpose: To investigate the dependence of the maximum power of codes on the number of digits and the minimum code distance; to find an approach to determine the optimal rules for constructing the check vector of a separable code from the point of view of ensuring minimal redundancy with a given reliability of message transmission. **Methods:** Computer simulation has been used to conduct experimental studies. For theoretical studies, the method of analytical review, graph theory, and coding theory have been applied. **Results:** Theoretical and experimental studies have obtained certain specific cases of power distribution for code alphabets with a given Hamming distance for various constant lengths, generated using the previously described algorithm. A method for doubling the power of arbitrary binary codes is proposed and described, as well as a method for obtaining codes with the least redundancy of powers $M=2^f$, where f is a natural number for a predetermined minimum code distance by recursively using the method proposed in the article. **Practical significance:** An algorithm has been developed for doubling the power of the code alphabet while maintaining the required reliability of data transmission. A technique for analyzing the resulting matrices of code vectors is obtained in order to determine the rules for calculating check bits without using cyclic algorithms.

Keywords: Anti-jamming coding, least redundant codes, code alphabet, code word, Hamming distance, separable codes.

References

1. Hamming R. W. Error detecting and error correcting codes. The Bell system technical journal. 1950, vol. 29, Iss. 2. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

2. Pronin G. Yu., Volkov E. A., Ivanov Yu. V. Algoritm generatsii uslovno optimal'nykh kodov [Algorithm for generating conditionally optimal codes]. *Transport: problemy, idei, perspektivy* [Transport: problems, ideas, prospects]. 2022, Iss. 2, pp. 96–98. (In Russian)

3. Manoj S. G., Mohan A. K., Sri Ganesh N. L. et al. Diagonal Hamming Based Multi-Bit Error Detection and Correction Technique for Memories. International Conference on Communication and Signal Processing. July 28–30, 2020. DOI: 10.1109/ICCSP48568.2020.9182249.

4. Van W. J., Lannoo J., Vankeirsbilck J. et al. Effectiveness of Hamming Single Error Correction Codes Under Harsh Electromagnetic Disturbances. International Symposium on Electromagnetic Compatibility. August 27–30, 2018. DOI: 10.1109/EMCEurope.2018.8485176.

5. Tshagharyan G., Hatutyunyan G., Shoukourian S. et al. Experimental Study on Hamming and Hsiao Codes in the Context of Embedded Applications. Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, September 29 — October 2, 2017.

6. Musayelyan R. Hamming Distance Based Data Correction Combined With Low Power XOR Circuit. Proceedings of 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021.

7. Sridevi N., Jamal K., Mannem K. Implementation of Error Correction Techniques in Memory Applications. 5th International Conference on Computing Methodologies and Communication. April 8–10, 2021. DOI: 10.1109/ICCMC51019.2021.9418432.

8. Tolentino L. K. S., Valenzuela I. C., Serfa R. O. Juan Overhead Interspersing of Redundancy Bits Reduction Algorithm by Enhanced Error Detection Correction Code. Journal of Engineering Science & Technology Review. 2019, vol. 12, Iss. 2.

9. Shahariar Parvez A. H. M. et al. Design and implementation of hamming encoder and decoder over FPGA. International Conference on Computer Networks and Communication Technologies: ICCNCT 2018. Springer Singapore, 2019, pp. 1005–1022.

10. Panem C., Gad V., Gad R. Polynomials in Error Detection and Correction in Data Communication System. Coding Theory, 2019, p. 29.

Received: April 14, 2023
Accepted: May 20, 2023

Author's information:
Anton. A. BLYUDOV — PhD in Engineering, Associate Professor; blyudov@pgups.ru
Dmitry V. PIVOVAROV — PhD in Engineering, Associate Professor; pivovarov.d.v.spb@gmail.com
Georgy Yu. PRONIN — Postgraduate Student; georgiy3pronin@gmail.com