УДК 519.725

ПРОСТОЙ АЛГОРИТМ ДЕКОДИРОВАНИЯ АЛГЕБРАИЧЕСКИХ КОДОВ

С. В. Федоренко,
канд. техн. наук, доцент
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассматривается алгоритм декодирования, который представляется самым простым и естественным в классе алгоритмов декодирования алгебраических кодов до их корректирующей способности. Приводятся вывод, описание, анализ и доказательство корректности алгоритма. Асимптотическая сложность алгоритма совпадает со сложностью лучших алгоритмов декодирования, а описание является самым простым из описаний известных алгоритмов.

Введение

Корректирующие коды получили широкое применение в задачах передачи, хранения и защиты информации. Для кодов, имеющих алгебраическую структуру, существуют алгебраические алгоритмы декодирования. Под алгебраическим декодированием понимают методы декодирования, основанные на алгебраических свойствах кодов и состоящие в решении уравнений и/или систем уравнений с полиномиальной сложностью. Классические алгебраические алгоритмы декодирования обеспечивают исправление ошибок до конструктивной корректирующей способности кода, но имеют довольно сложное и громоздкое описание.

В работах [7, 8] предложен алгоритм декодирования, который представляется самым простым и естественным в классе алгоритмов декодирования алгебраических кодов до их корректирующей способности. Асимптотическая сложность алгоритма совпадает со сложностью лучших алгоритмов декодирования, а описание является самым простым из описаний известных алгоритмов.

В статье предлагается описание, оригинальный вывод и доказательство корректности алгоритма. Приводится пример алгоритма декодирования.

Основные понятия и определения

Описание алгоритма декодирования приведем для кодов Рида—Соломона, так как для других классов алгебраических кодов принципиальных отличий в описании алгоритма нет.

Рассмотрим код Рида—Соломона (n, k, d) над конечным полем $\mathrm{GF}(q)$ с длиной $n = q - 1$, числом информационных символов k и конструктивным расстоянием $d = n - k + 1$, где q — степень простого числа. Корректирующая способность кода равна $\left[\frac{d-1}{2}\right]$, где $[a]$ — целая часть числа a.

Порождающий многочлен кода Рида—Соломона обозначим через

$$g(x) = \prod_{i=b}^{b+d-2} (x - \alpha^i),$$

где b — произвольное натуральное число; α — примитивный элемент $\mathrm{GF}(q)$.

Далее для упрощения изложения будем рассматривать только случай $b = 1$.

Принятый вектор представлен многочленом

$$R(x) = \sum_{i=0}^{n-1} r_i x^i = C(x) + E(x) = \sum_{i=0}^{n-1} c_i x^i + \sum_{i=0}^{n-1} e_i x^i,$$

где $C(x)$ — кодовое слово; $E(x)$ — вектор ошибок.

Пусть $E(x)$ содержит $t \le \frac{d-1}{2}$ ошибок, которые имеют координаты $i_1, i_2, \ldots, i_t$, причем $0 \le i_1 < i_2 < \ldots < i_t \le n - 1$, и значения $e_{i_1}, e_{i_2}, \ldots, e_{i_t}$. Назовем величины $Z_1 = \alpha^{i_1}, Z_2 = \alpha^{i_2}, \ldots, Z_t = \alpha^{i_t}$ локаторами ошибок, а $Y_1 = e_{i_1}, Y_2 = e_{i_2}, \ldots, Y_t = e_{i_t}$ — значениями ошибок.

Многочлен локаторов ошибок обозначим через

$$W(x) = \prod_{i=1}^{t} (x - Z_i),$$

где Z_i — локатор ошибки в векторе ошибок $E(x)$. Положим по определению $W(x) = 1$, если ошибок нет.

Известно несколько методов кодирования для кодов Рида—Соломона. В настоящей работе при-

меняется спектральное кодирование в частотной области для несистематического кодирования. Заметим, что алгоритм декодирования не зависит от метода кодирования. Информационный многочлен кода Рида—Соломона обозначим как

$$M(x)=\sum_{i=0}^{k-1} m_i x^i.$$

При спектральном кодировании компонента c_i кодового слова $C(x)$ вычисляется как

$$c_i = M(\alpha^i),\ i\in[0, n-1].$$

Вывод алгоритма декодирования

Вывод алгоритма декодирования, основанный на интерпретации работ [1–4], предложен в работах [5, 6].

Если $r_i = c_i$, то $r_i = M(\alpha^i)$. Если $r_i \neq c_i$, то $W(\alpha^i) = 0$. Следовательно:

$$W(\alpha^i)r_i = W(\alpha^i)M(\alpha^i),\ i\in[0, n-1].$$

Пусть $P(x) = W(x)M(x)$. Тогда ключевое уравнение имеет вид

$$W(\alpha^i)r_i = P(\alpha^i),\ i\in[0, n-1].$$

Построим такой интерполяционный многочлен $T(x)$, что

$$T(\alpha^i) = r_i,\ i\in[0, n-1],$$

где $\deg T(x) < n$.

Далее, из

$$W(\alpha^i)T(\alpha^i) = P(\alpha^i),\ i\in[0, n-1]$$

имеем сравнение

$$\begin{cases} W(x)T(x)\equiv P(x) \bmod x^n - 1 \\ \deg W(x)\le \dfrac{d-1}{2} \\ \text{maximize } \deg W(x) \end{cases}.$$

Учитывая, что $\deg P(x) = \deg M(x) + \deg W(x) \le$ $\le (k-1) + \dfrac{d-1}{2} \le \dfrac{n+k}{2} - 1 < \dfrac{n+k}{2}$, переписываем условие решения сравнения на эквивалентное

$$\begin{cases} W(x)T(x)\equiv P(x) \bmod x^n - 1 \\ \deg P(x) < \dfrac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases}.$$

Решаем сравнение применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$, получая многочлены $P(x)$ и $W(x)$. Информационный многочлен получается делением

$$M(x) = \frac{P(x)}{W(x)}.$$

Алгоритм декодирования

Приведем алгоритм из работы [7]. Заметим, что этот алгоритм ранее был введен в работе [8]. Для упрощения изложения будем рассматривать только классические коды Рида—Соломона с параметрами $n = q - 1$ и $b = 1$.

1. Интерполяция.

Построим такой интерполяционный многочлен $T(x)$, что

$$T(\alpha^i) = r_i,\ i\in[0, n-1],$$

где $\deg T(x) < n$.

2. Незаконченное вычисление наибольшего общего делителя.

Решаем сравнение

$$\begin{cases} W(x)T(x)\equiv P(x) \bmod x^n - 1 \\ \deg P(x) < \dfrac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases}$$

применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$, получая единственную пару многочленов $P(x)$ и $W(x)$.

3. Деление.

Информационный многочлен есть

$$M(x) = \frac{P(x)}{W(x)}.$$

Асимптотическая сложность алгоритма $O(n(\log n)^2)$ совпадает со сложностью лучших классических алгоритмов декодирования кодов Рида—Соломона [9–11].

Первый шаг алгоритма может быть выполнен любым быстрым алгоритмом вычисления дискретного преобразования Фурье над конечным полем, например [12, 13], со сложностью $O(n(\log n)^2)$ операций. В случае, когда необходимо минимизировать число умножений, лучший алгоритм для малых длин ($n \le 511$) предложен в работе [14].

Одна из лучших реализаций второго шага есть алгоритм Мёнка [15] со сложностью $O(n(\log n)^2)$ операций, который также воспроизведен в монографиях [16, 17]. Заметим, что при этом второй шаг полностью совпадает с алгоритмом решения ключевого уравнения Сугиямы и других [18].

Деление на третьем шаге алгоритма выполняется за $O(n\log n\log\log n)$ операций.

При приложении алгоритма декодирования к другим классам алгебраических кодов, таких как коды Боуза—Чоудхури—Хоквингема, коды Гоппы или альтернантные коды, необходимо добавить дополнительный шаг, восстанавливающий кодовое слово по информационному многочлену.

Корректность алгоритма декодирования

Для завершения вывода алгоритма декодирования необходимо доказать существование и единственность полученного решения.

Теорема. *Алгоритм декодирования приводит к единственному решению при декодировании до корректирующей способности кода Рида—Соломона.*

Доказательство. Пусть имеются два решения сравнения

$$\begin{cases} W(x)T(x) \equiv P(x) \bmod x^n - 1 \\ \deg P(x) < \dfrac{n+k}{2} \\ \text{maximize} \deg P(x) \end{cases}. \qquad (*)$$

Первое решение, полученное применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$ с правилом остановки $\deg P(x) < \frac{n+k}{2}$, дает пару многочленов $P(x)$ и $W(x)$. Если $P(x)$ делится на $W(x)$ без остатка, то многочлен $M(x) = \dfrac{P(x)}{W(x)}$ будет информационным многочленом кода Рида—Соломона.

Заметим, что расширенный алгоритм Евклида надо проводить с нулевого (фиктивного) шага, а не с первого, как обычно. Заканчивать расширенный алгоритм Евклида надо дополнительным (с нулевым остатком) шагом.

Другое решение (истинное) удовлетворяет сравнению

$$\widetilde{W}(x)T(x) \equiv \widetilde{P}(x) \bmod x^n - 1$$

при декодировании до корректирующей способности кода Рида—Соломона $\deg \widetilde{W}(x) \le \frac{d-1}{2}$ и приводит к другому информационному многочлену кода Рида—Соломона

$$\widetilde{M}(x) = \frac{\widetilde{P}(x)}{\widetilde{W}(x)}.$$

Вначале рассмотрим вырожденный случай, когда решение сравнения (*) заканчивается на нулевом шаге расширенного алгоритма Евклида, и докажем, что информационные многочлены, полученные из обоих решений, $M(x) = \widetilde{M}(x)$, совпадают.

Пусть $\widetilde{W}(x) = 1$. Тогда $\widetilde{M}(x) = M(x) = P(x) = T(x)$, так как из нулевого шага расширенного алгоритма Евклида следует, что $\deg T(x) \le k - 1 < \frac{n+k}{2}$.

Далее будем полагать, что $\widetilde{W}(x) \ne 1$.

Заметим, что $\deg W(x) \le n - \frac{n+k}{2} = \frac{d-1}{2}$ по свойству степени линейных коэффициентов в расширенном алгоритме Евклида.

Выполним деление с остатком

$$P(x) = W(x)M(x) + U(x),$$

причем $\deg U(x) < \deg W(x)$.

Если остаток от деления равен нулю: $U(x) = 0$, то многочлен $M(x)$ будет информационным многочленом кода Рида—Соломона.

Очевидно, что $\deg \widetilde{P}(x) = \deg \widetilde{M}(x) + \deg \widetilde{W}(x) \le$

$$\le (k-1) + \frac{d-1}{2} \le \frac{n+k}{2} - 1 < \frac{n+k}{2}.$$

Покажем, что информационные многочлены, полученные из обоих решений, совпадают, т. е.

$$\begin{cases} M(x) = \widetilde{M}(x) \\ U(x) = 0 \end{cases}.$$

После преобразований

$$P(x)\widetilde{W}(x) \equiv \left(W(x)T(x)\right)\widetilde{W}(x) =$$

$$= W(x)\left(T(x)\widetilde{W}(x)\right) \equiv W(x)\widetilde{P}(x) \bmod x^n - 1$$

получаем сравнение

$$P(x)\widetilde{W}(x) \equiv W(x)\widetilde{P}(x) \bmod x^n - 1.$$

Оценим степени произведения многочленов в каждой части сравнения. Из

$$\begin{cases} \deg W(x),\ \widetilde{W}(x) \le \dfrac{d-1}{2} \\ \deg P(x),\ \widetilde{P}(x) < \dfrac{n+k}{2} \end{cases}$$

следует

$$\begin{cases} \deg\left(P(x)\widetilde{W}(x)\right) < \dfrac{n+k}{2} + \dfrac{d-1}{2} = n \\ \deg\left(W(x)\widetilde{P}(x)\right) < \dfrac{d-1}{2} + \dfrac{n+k}{2} = n \end{cases},$$

т. е.

$$\begin{cases} \deg\left(P(x)\widetilde{W}(x)\right) < n \\ \deg\left(W(x)\widetilde{P}(x)\right) < n \end{cases}.$$

Видно, что степени произведения многочленов в каждой части сравнения не превышают степень модуля $x^n - 1$, т. е. сравнение справедливо как равенство для многочленов

$$P(x)\widetilde{W}(x) = W(x)\widetilde{P}(x).$$

Отсюда после деления имеем

$$P(x) = W(x)\frac{\widetilde{P}(x)}{\widetilde{W}(x)} = W(x)\widetilde{M}(x).$$

Из последнего равенства видно, что $P(x)$ делится на $W(x)$ без остатка, т. е. $U(x) = 0$ и $M(x) = \dfrac{P(x)}{W(x)}$.

После еще одного деления имеем

$$\frac{P(x)}{W(x)} = \widetilde{M}(x)$$

и

$$M(x) = \widetilde{M}(x).$$

Показано, что решение сравнения (*), полученное применением расширенного алгоритма Евклида, всегда существует и совпадает с истинным решением. Ч. т. д.

Пример

Рассмотрим процедуры кодирования и декодирования для кода Рида—Соломона (4, 2, 3). Коэффициенты многочлена будем записывать как вектор-строку и как вектор-столбец (без обозначения операции транспонирования).

Конечное поле. Вначале введем конечное поле GF(5) = {0, 1, 2, 3, 4}mod5. Или в другом представлении: $GF(5)=\{0, \alpha^0, \alpha^1, \alpha^2, \alpha^3\}$, где $\alpha=2$ — примитивный элемент:

$$\begin{array}{cccc|c} \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4=1 \\ \hline 1 & 2 & 4 & 3 & 1 \end{array}.$$

Код Рида—Соломона. Введем код Рида—Соломона (4, 2, 3) над GF(5). Его порождающий многочлен имеет вид

$$g(x)=(x-\alpha)(x-\alpha^2)=$$
$$=(x-2)(x-4)=x^2+4x+3,$$

порождающая матрица

$$G=\begin{bmatrix} g_0 & g_1 & g_2 & 0 \\ 0 & g_0 & g_1 & g_2 \end{bmatrix}=\begin{bmatrix} 3 & 4 & 1 & 0 \\ 0 & 3 & 4 & 1 \end{bmatrix}$$

и проверочная матрица

$$H=\begin{bmatrix} \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \end{bmatrix}=\begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \end{bmatrix}.$$

Спектральное кодирование. Кодовое слово вычисляется по формуле

$$\mathbf{C}=V\mathbf{M},$$

где матрица Вандермонда есть

$$V=\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^3 & \alpha^2 & \alpha^1 \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix}.$$

Пусть информационный многочлен есть

$$M(x)=\begin{bmatrix} m_0 & m_1 & 0 & 0 \end{bmatrix}=$$
$$=\begin{bmatrix} 2 & 3 & 0 & 0 \end{bmatrix}=2+3x.$$

Тогда процедура кодирования имеет вид

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}=\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^3 & \alpha^2 & \alpha^1 \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ 0 \\ 0 \end{bmatrix};$$

$$\begin{bmatrix} 0 \\ 3 \\ 4 \\ 1 \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix};$$

$$C(x)=\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}=$$
$$=\begin{bmatrix} 0 & 3 & 4 & 1 \end{bmatrix}=3x+4x^2+x^3.$$

Обратное преобразование. Запишем матрицу, обратную к матрице Вандермонда:

$$V^{-1}=-\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix}.$$

Тогда обратное дискретное преобразование Фурье есть

$$\mathbf{M}=V^{-1}\mathbf{C};$$

$$C(x)=\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}=$$
$$=\begin{bmatrix} 0 & 3 & 4 & 1 \end{bmatrix}=3x+4x^2+x^3;$$

$$\begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix}=-\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix}\begin{bmatrix} 0 \\ 3 \\ 4 \\ 1 \end{bmatrix};$$

$$M(x)=\begin{bmatrix} m_0 & m_1 & 0 & 0 \end{bmatrix}=$$
$$=\begin{bmatrix} 2 & 3 & 0 & 0 \end{bmatrix}=2+3x.$$

Декодирование. Пусть из канала принят вектор $R(x) = C(x) + E(x)$:

$$\begin{array}{lcccc} C(x) & = & (0 & 3 & 4 & 1) \\ E(x) & = & (0 & 0 & 2 & 0) \\ R(x) & = & (0 & 3 & 1 & 1) \\ \hline Z & & \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ Z & & 1 & 2 & 4 & 3 \end{array}$$

$$R(x)=\begin{bmatrix} r_0 & r_1 & r_2 & r_3 \end{bmatrix}=$$
$$=\begin{bmatrix} 0 & 3 & 1 & 1 \end{bmatrix}=3x+x^2+x^3.$$

1. Интерполяция:

$$\begin{bmatrix} 0 \\ 0 \\ 3 \\ 2 \end{bmatrix}=-\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix}\begin{bmatrix} 0 \\ 3 \\ 1 \\ 1 \end{bmatrix};$$

$$\mathbf{T} = V^{-1}\mathbf{R} = \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \end{bmatrix} = \\ = \begin{bmatrix} 0 & 0 & 3 & 2 \end{bmatrix} = 3x^2 + 2x^3.$$

2. Незаконченное вычисление наибольшего общего делителя:

$$\begin{cases} W(x)T(x) \equiv P(x) \bmod x^n - 1 \\ \deg P(x) < \dfrac{n+k}{2} \\ \text{maximize} \deg P(x) \end{cases};$$

$$\begin{cases} W(x)\left(3x^2 + 2x^3\right) \equiv P(x) \bmod x^4 - 1 \\ \deg P(x) < 3 \end{cases};$$

$$x^4 - 1 = \left(3x^2 + 2x^3\right)(3x+3) + \left(x^2 - 1\right);$$

$$(3x+3)\left(3x^2 + 2x^3\right) = -\left(x^2 - 1\right) + x^4 - 1;$$

$$(3x+3)\left(3x^2 + 2x^3\right) \equiv 4x^2 + 1 \bmod x^4 - 1;$$

$$\begin{cases} W(x) = 3x + 3 = 3(x-4) = 3\left(x - \alpha^2\right) \\ P(x) = 4x^2 + 1 \end{cases}.$$

3. Деление:

$$M(x) = \frac{P(x)}{W(x)} = \frac{4x^2 + 1}{3x + 3} = 3x + 2.$$

Информационный многочлен есть

$$M(x) = \begin{bmatrix} m_0 & m_1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 0 & 0 \end{bmatrix} = 2 + 3x.$$

Заключение

Рассмотренный метод декодирования предложен Гао [7] и Шиозаки [8], однако вывод и анализ алгоритма, доказательство его корректности и генетическая связь с алгоритмами [1–4, 9, 10, 18] являются оригинальным результатом автора [5, 6].

На текущий момент этот метод декодирования и его модификации представляются автору самыми простыми для кодов с ограниченной длиной при любых реализациях.

Автор выражает признательность фонду имени Александра фон Гумбольдта (Германия) за многолетнюю поддержку научных исследований.

Литература

1. **Welch L., Berlekamp E. R.** Error correction for algebraic block codes. U.S. Patent 4,633,470. Sep. 27, 1983.
2. **Morii M., Kasahara M.** Generalized key-equation of remainder decoding algorithm for Reed—Solomon codes // IEEE Transactions on Information Theory. Nov. 1992. Vol. IT-38. N 6. P. 1801–1807.
3. **Chambers W. G.** Solution of Welch—Berlekamp key equation by Euclidean algorithm // Electronics Letters. 1993. Vol. 29. N 11. P. 1031.
4. **Gemmell P., Sudan M.** Highly resilient correctors for polynomials // Information Processing Letters. 1992. Vol. 43. N 4. P. 169–174.
5. **Fedorenko S. V.** A simple algorithm for decoding Reed—Solomon codes and its relation to the Welch—Berlekamp algorithm // IEEE Transactions on Information Theory. Mar. 2005. Vol. IT-51. N 3. P. 1196–1198.
6. **Fedorenko S. V.** Correction to «A simple algorithm for decoding Reed—Solomon codes and its relation to the Welch—Berlekamp algorithm» // IEEE Transactions on Information Theory. Mar. 2006. Vol. IT-52. N 3. P. 1278.
7. **Gao S.** A new algorithm for decoding Reed—Solomon codes // Communications, Information and Network Security / V. Bhargava, H. V. Poor, V. Tarokh, and S. Yoon, Eds. Norwell, MA: Kluwer, 2003. Vol. 712. P. 55–68.
8. **Shiozaki A.** Decoding of redundant residue polynomial codes using Euclid's algorithm // IEEE Transactions on Information Theory. Sep. 1988. Vol. IT-34. N. 5. P. 1351–1354.
9. **Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.** Теория кодов, исправляющих ошибки. М.: Связь, 1979. 744 с.
10. **Блейхут Р.** Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986. 576 с.
11. **Justesen J.** On the complexity of decoding Reed—Solomon codes // IEEE Transactions on Information Theory. Mar. 1976. Vol. IT-22. N 2. P. 237–238.
12. **Wang Y., Zhu X.** A fast algorithm for the Fourier transform over finite fields and its VLSI implementation // IEEE Journal on Selected Areas in Communications. Apr. 1988. Vol. 6. N 3. P. 572–577.
13. **Afanasyev V.** On complexity of FFT over finite field: Proc. of the Sixth Joint Swedish-Russian International Workshop on Information Theory. Molle, Sweden, August 1993. P. 315–319.
14. **Трифонов П. В., Федоренко С. В.** Метод быстрого вычисления преобразования Фурье над конечным полем // Проблемы передачи информации. 2003. Т. 39. Вып. 3. С. 3–10.
15. **Moenck R. T.** Fast computation of GCDs: Proc. of the 5th Annual ACM Symposium on Theory of Computing. Austin, TX, 1973. P. 142–151.
16. **Ахо А., Хопкрофт Дж., Ульман Дж.** Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 535 с.
17. **Блейхут Р.** Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989. 448 с.
18. **Sugiyama Y., Kasahara M., Hirasawa S., Namekawa T.** A method for solving key equation for decoding Goppa codes // Information and Control. 1975. Vol. 27. P. 87–99.