Изв. АН Эстонии. Физ. Матем., 1990, 39, № 1, 26—31
https://doi.org/10.3176/phys.math.1990.1.04

УДК 681.3.06

Виктория ЧЕРНЫШЕВА

ОЦЕНКА ПРОГРАММНОЙ РЕАЛИЗАЦИИ МЕТОДОВ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ

(Представил Н. Алумяэ)

1. Введение

В настоящее время методы кодирования и декодирования в основном рассматриваются с традиционных позиций теории кодирования. Специфика вычислительных сетей приводит к постановке новых задач. В первую очередь главное внимание уделяется методам помехоустойчивого кодирования, обладающим простой программной реализацией. Это связано с широким использованием в сетях программируемых устройств, в том числе микропроцессоров. Во-вторых, возрастает внимание к вычислительным сетям, использующим метод адаптивной коммутации. Разрабатываются процедуры доступа к сети адаптивной коммутации, базирующиеся на расширении стандартных протоколов сетей с коммутацией пакетов. Это выдвигает на первый план вопросы сравнения существующих методов защиты от ошибок. Актуальным остается вопрос выбора и разработки показателей оценки методов, позволяющих определять сложность их реализации и производить сопоставительный анализ методов.

При инженерном подходе к исследованию программной реализации процедуры кодирования и декодирования первостепенными и общими характеристиками, определяющими затраты реализации данного метода, являются объем памяти и время вычисления. Эти параметры зависят от используемого языка программирования, типа ЭВМ и, в конечном счете, от параметров алгоритма. Один из способов оценки сложности методов кодирования/декодирования базируется на теории вероятностных автоматов [1–3]. Авторами [1–3] рассматривается проблема выбора кода с заданным методом декодирования и под временем вычисления понимается число тактов решения задачи на машине Тьюринга.

В процессе передачи данных под временем решения задачи правомерно понимать число преобразующих операций. Фактически оно определяется суммарным числом умножений и сложений, что позволяет отразить тип применяемой ЭВМ. Таким образом, мы предлагаем оценивать сложность программной реализации методов кодирования и декодирования следующими характеристиками: длиной алгоритма и временем вычисления по алгоритму. Длина алгоритма оценивается [2, 3] логической сложностью функции кодирования $f=\varphi_n$ или декодирования $f=\psi_n$, равной числу содержащихся в двоичном описании схемы функциональных элементов (ФЭ)

$$\varkappa_S(f) = \min_{S \in G_k(f)} \varkappa(S),$$

где $G_k^n(f)$ — множество всех схем, реализующих функцию f. Время вычисления определяется числом преобразующих операций $W_0(K)$, $W_0(D)$ соответственно при кодировании и декодировании [4, 5].

2. Оценки сложности линейных кодов

Разработанные в [1–5] критерии оценки сложности двоичных линейных кодов удобно обобщить по классам кодов и представить в виде следующей таблицы.

Класс кодов	$\varkappa(\varphi_n)$	$W_0(K)$
Блоковые линейные систематические	rk	$2rk$
Блоковые линейные несистематические	kn	$2kn$
Циклические	k	$2kr$
Древовидные, описанные с помощью матриц	$1/2\,k_e(n_e+n_0)$	$k_e+(n_e+n_0)$
Несистематические сверточные	$1/2\,k_e(n_e+n_0)$	$k_e n_0(2m-m_e+1)$
Каскадные первого порядка	$cn\log^4 n$	$2k_a\{k_b n_b-k_b^2+n_a\}$

Здесь k, r — числа информационных и проверочных символов кодовой комбинации длины n; (n_e, k_e), (n_0, k_0) — параметры древовидных кодов; (n_a, k_a), (n_b, k_b) — параметры внутреннего и внешнего кодов; $n=n_a n_b$; c — константа.

3. Оценка сложности методов декодирования

Сложность и разнообразие математического аппарата методов декодирования затрудняет разбиение их на классы. Поэтому среди множества методов выделим классические, а также перспективные в плане быстродействия.

3.1. Для **синдромного декодирования** [6–8] получены следующие оценки: в режиме обнаружения $\varkappa(\psi_n)\leqslant nr$ [2, 3], $W_0(D)\leqslant 2\,nr$ [4], в режиме исправления $\varkappa(\psi_n)=n2^r$ [3], $W_0(D)\leqslant 2nr+r2^r$ [4], для кода Хэмминга, исправляющего одну ошибку $W_0(D)\leqslant 3rn$.

3.2. Оценим сложность программной реализации декодирования по **алгоритму Прейнджа** [6–9] **групповых кодов.**

Пусть N_p — число перестановок или, точнее, число матриц перестановок π_i размерности $n\times n$, $i=1, 2, \ldots$, $\pi_i\in P$. Для каждой перестановки $\pi_i\in P$ находится кодовый вектор $L_y(\pi_i b)$. Это требует не более $2kr$ шагов для реализации операции кодирования $L_y(b)=(b\vdots b_k H_1^T)$, $H=[H_1\vdots I]$, H_1 — матрица размерности $r\times k$. Для осуществления перестановки $\pi_i b$ необходимо $n\times n$ шагов. Вычисление расстояния и сравнение его с величиной $(d_0-1)/2$ требует порядка $(2n+2\log_2 n)$ шагов. Сложность декодера в целом для N_p перестановок определяется как

$$W_0(D)\approx N_p(2kr+n^2+2n+2\log_2 n)\leqslant N_p(2kr+n^2).$$

Согласно [2], операция кодирования линейного кода осуществляется схемой сложности, определяемой размерами матрицы. Схема, реализующая перестановку, не содержит ФЭ, поэтому

$$\varkappa(\psi_n)\leqslant rk.$$

Для **циклических кодов** перестановка осуществляется путем циклического сдвига полученной кодовой комбинации. Логическая сложность алгоритма не изменяется. Так как число сдвигов не превышает числа информационных символов, то

$$W_0(D)\leqslant k(2kr+r+2\log_2 n)\leqslant kr(2k+1).$$

3.3. Сложность декодирования по **алгоритму Меггита** циклических кодов [6, 8, 10] зависит от числа сравнений синдрома полученной кодовой комбинации. Для вычисления синдрома необходимо $2kr$ преобразующих операций. Проверка на совпадение с одним из синдромов длины r символов множества синдромов осуществляется не более B раз при каждом из n сдвигов. Наблюдается закономерность, для которой

$$B=\frac{(n-1)!}{(t-1)!(n-t)!}+1.$$

Если $t=1$, то $B=1$. Зная величину B, оценим сложность декодера

$$W_0(D)\leqslant 2kr+Brn.$$

Аналогично синдромному декодированию вычисленный схемой сложности rk-синдром поступает в логическую схему нахождения исправляемой конфигурации ошибки. Число подсхем равно B и

$$\varkappa(\psi_n)\leqslant rk+Br=r(k+B).$$

3.4. Модификация декодера Меггита — т. н. **декодер с вылавливанием ошибок** — применяется к некоторым циклическим кодам, исправляющим близко расположенные ошибки или пакеты ошибок. Поскольку ненулевые символы комбинации ошибок появляются в соответствующих разрядах синдрома, то сложность построенного декодера уменьшается и оценки приобретают следующие значения

$$\varkappa(\psi_n)\leqslant rk+r=r(k+1),\quad W_0(D)\leqslant 2kr+ntr.$$

3.5. Сложность реализации **мажоритарного декодера** с разделенными проверками [7, 8] для **групповых кодов** определяется размерностью проверочных матриц Hm, $m=\overline{1,k}$. Согласно [4], $W_0(D)\leqslant 2kJ(n-1)$, где J — число нетривиальных проверок. Оценим логическую сложность. Поскольку алгоритм представляется цепью подсхем, количество которых равно числу информационных символов и каждая имеет сложность Jn, то сложность декодера определяется как

$$\varkappa(\psi_n)=\sum_{m=1}^{k}Jn\leqslant Jkn.$$

При декодировании **циклического кода** используется проверочная матрица, выбранная для первого символа. Сдвиг матрицы осуществляется регистром сдвига и учитывается при вычислении автоматной сложности. Поэтому $\varkappa(\psi_n)\leqslant Jn$.

3.6. Оценим сложность декодирования сверточных кодов по **алгоритму Витерби** [6, 8, 11]. Для декодирования одного информационного блока длиной k_0 символов на первом этапе требуется

$$m_e n_0 2^{k_e+1}+(2^{k_e}-1)2^{k_e-1}(m_e-1)k_0$$

операций для выбора выживших путей. На последующия этапах —

$$n_0 2^{k_e}+(2^{k_e}-1)2^{k_e-1}(m_e-1)k_0$$

операций. Для определения очередных кодовых символов c_{ij} при удлинении пути по 2^{ke} информационным символам a_i, учитывая, что первые $n_e 2^{ke}$ кодовых и $k_e 2^{ke}$ информационных символов заданы, необходимо $k_e 2^{ke}$ операций.

$$W_0(D)=2^{k_e+1}n_e+2^{2k_e-1}(m_e-1)k_0-2^{k_e-1}(m_e-1)k_0+$$
$$+(m-m_e)[n_0 2^{k_e}+2^{2k_e-1}(m_e-1)k_0-2^{k_e-1}\times$$
$$\times(m_e-1)k_0+k_e 2^{k_e}]\leqslant S_1 2^{k_e}+S_2 2^{2k_e-1},$$

$$S_1=2n+(k_e+k_0)(m-m_e),\quad S_2=(k_e-k_0)(m-m_e+1).$$

Логическая сложность алгоритма складывается из следующих компонентов. Расстояние вычисляется на первом этапе схемой сложности $2m_e n_0$, на последующих — $2n_0$. Выбор выживших путей можно выполнить схемой сложности не более

$$(2^{k_e+1}-1)k_e+2^{k_e-1}\log n_e.$$

Согласно [2] схема сравнения по сложности пропорциональна $\log n_e$. Общая логическая сложность алгоритма

$$\varkappa(\psi_n)\leqslant 2m_e n_0+2n_0+(2^{k_e+1}-1)k_e+2^{k_e-1}\log n_e+2^{k_e}k_e+2^{k_e}\log n_e\leqslant$$
$$\leqslant 2n_0(m_e+1)+2^{k_e}(\log n_e+3k_e).$$

3.7. Преобразование алгоритма декодирования Питерсона—Горстейна—Циглера циклических кодов сводится Э. Р. Берлекэмпом в вариации Месси (**алгоритм Берлекэмпа—Месси** [6, 10]) к нахождению и способу построения регистра минимальной длины $(L_r, \Lambda^r(x))$, генерирующего последовательность $s_1, s_2, \ldots, s_r$. В каждой итерации требуется для умножения матриц

$$\begin{bmatrix} \Lambda^{(r)}(x) \\ B^{(r)}(x) \end{bmatrix}=\begin{bmatrix} 1 & -\Delta_r x \\ \Delta_r^{-1}\delta r & (1-\delta r)x \end{bmatrix}\begin{bmatrix} \Lambda^{(r-1)}(x) \\ B^{(r-1)}(x) \end{bmatrix}$$

не более $2t$ умножений, для вычисления Δ_r не более t умножений [10]. Согласно [2], операции в поле $GF(2^m)$ выполняются логической схемой сложности: 1) для сложения двух элементов поля $\varkappa=m$; 2) для умножения произвольного элемента на примитивный элемент поля $\varkappa\leqslant 3m$; 3) для умножения двух элементов поля $\varkappa\leqslant 8m$. Тогда, зная, что вычисление синдромов s_j, $j=\overline{1,2t}$ требует t умножений и $(t-1)$ сложений в поле $GF(2^m)$, имеем

$$\varkappa(S_1)=t8m+(t-1)m\leqslant m(9t-1),$$
$$W_0(S_1)=2tmt+2tm(t-1)=2tm(2t-1).$$

Вычисление Δ_r выполняется схемой сложности

$$\varkappa(S_2)=t8m+tm=9tm,$$
$$W_0(S_2)=t^2m+mt.$$

Вычисление промежуточных многочленов осуществляется схемами сложности

$$\varkappa(S_3)=(t-1)8m+(t-1)m+8mt=17mt-9m,$$
$$W_0(S_3)\leqslant 6t^2m-2tm.$$

Нахождение корней многочлена $\Lambda(x)$ выполняется процедурой Ченя, реализуемой схемой сложности

$$\varkappa(S_4)=tm,\qquad W_0(S_4)=2t^2m(2^m-2).$$

Общая сложность декодера оценивается

$$\varkappa(\psi_n)=\sum_{i=1}^{4}\varkappa(S_i)\leqslant 2m(18t-5),$$
$$W_0(D)\leqslant 7t^2m-3tm+t^2m2^{m+1}.$$

3.8. Одним из быстрых алгоритмов декодирования является **рекуррентный алгоритм Берлекэмпа—Месси.** В нем используются алгоритмы свертки, что уменьшает число умножений. Р. Блейхут [10] оценивает сложность алгоритма только числом умножений

$$M=0(n\log n\,2^{\log^* n}),$$

где $\log^* n$ — число, показывающее, сколько раз надо последовательно

проинтегрировать логарифмы по основанию 2 начиная с n, чтобы получить число, не превосходящее единицы. Определение числа сложения начнем с нахождения требуемого количества сложений в алгоритмах свертки.

Для алгоритмов свертки с простым $N=3, 5, 7$ наблюдается следующая закономерность (M, A — числа умножений и сложений)

$$M_N=1+M_{N-1}, \quad A=4(N-1)+A_{N-1}.$$

Зная, что если $N=2$, то $M_2=3$, $A_2=3$, получим

$$M_3=4, \quad A_3=11, \quad M_5=10, \quad A_5=31, \quad M_7=12, \quad A_7=58.$$

При

$$N=N_1N_2\dots N_d,$$

$$M=M_1M_2\dots M_d.$$

$$A=A_1N_2\dots N_d+M_1A_2N_3\dots N_d+\dots+M_1M_2\dots M_{d-1}A_d.$$

Исходя из этого, для алгоритмов свертки, представленных в [10], соответствующие числа сложений равны:

N	9	10	14	15	27	35	45	63	105	315	630
M	19	30	39	40	76	130	190	247	520	2470	7410
A	74	92	155	203	431	693	959	1620	3197	15757	42876

При представлении N степенью 2 ($N=2^s$) $M=N\log N$. На каждом уровне итерации рекуррентного алгоритма Берлекэмпа—Месси число обращений к уровню равно 2^l, $l=0, 1, \dots, 2^\rho$. Величина N равна $2^{-l}t$, т. е. на нулевом уровне $N=t$, на первом уровне $N=t/2$ и т. д. Число вычислений свертки на каждом уровне составляет $2t\times 2^l/\tau$, t/τ — число ветвей итерации. Определим число сложений в алгоритме:

$$A=\frac{2t}{\tau}A_t+\frac{2t}{\tau}\times 2\times A_{t/2}+\frac{2t}{\tau}\times 4\times A_{t/4}+\dots+\frac{t}{\tau}\times\frac{t}{2}\times A_1=$$

$$=\frac{3t^2}{4\tau}\{\log t(\log t-1)(\log t+1)+2\},$$

$$M=\frac{t^2}{\tau}\log t(\log t+1), \quad A\approx 0{,}75(\log t-1)M.$$

Общее число преобразующих операций равно

$$W_0(D)\leqslant t^2m2^{m+1}-2tm+n(\log n)2^{2\log^* n}.$$

Алгоритм свертки для многочленов степени не более t реализуется схемой сложности

$$\varkappa(s_5)=(t+1)^23m+2(t+1)m=3mt^2+8tm+5m.$$

Вычисление синдрома и нахождение корней многочлена локаторов ошибок происходит аналогично алгоритму Берлекэмпа—Месси.

$$\varkappa(\psi_n)=\varkappa(s_1)+\varkappa(s_5)+\varkappa(s_4)\leqslant 3mt^2+16tm+4m.$$

4. Заключение

Разработанные критерии оценки методов кодирования и декодирования позволяют определять вычислительную сложность существую-

щих и разрабатываемых методов. На основе полученных показателей произведен сравнительный анализ и определены параметры областей совместного применения методов кодирования и декодирования. При исследовании системы передачи данных с различными протоколами обмена показатели $W_0(K)$, $W_0(D)$ служат основой определения временной характеристики.

ЛИТЕРАТУРА

1. *Бассалыго Л. А., Зяблов В. В., Пинскер М. С.* // Пробл. передачи информ., 1977, № 3, 5—17.
2. *Блох Э. Л., Зяблов В. В.* Обобщенные каскадные коды. М., Связь, 1976.
3. *Блох Э. Л., Зяблов В. В.* Линейные каскадные коды. М., Наука, 1982.
4. *Чернышева В. А.* // Изв. АН ЭстССР. Физ. Матем., 1983, 32, № 3, 263—267.
5. *Чернышева В. А.* // Методы оптимизации сложных систем. М., Наука, 1987, 150—154.
6. *Кларк Д.*, мл., *Кейн Д.* Кодирование с исправлением ошибок в системах цифровой связи. М., Радио и связь, 1987.
7. *Колесник В. Д., Мирончиков Е. Т.* Декодирование циклических кодов. М., Связь, 1968.
8. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М., Мир, 1976.
9. *Зайдлер Е.* Системы передачи дискретной информации. М., Связь, 1977.
10. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М., Мир, 1986.
11. *Витерби А. Д., Омура Дж. К.* Принципы цифровой связи и кодирования. М., Радио и связь, 1982.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
26/V 1989

Viktoria TSERNÕŠEVA

KODEERIMIS- JA DEKODEERIMISMEETODITE PROGRAMSE REALISATSIOONI HINDAMINE

Artiklis on esitatud kooderi/dekooderi programse realisatsiooni hindamise kriteeriumid: algoritmi pikkus (funktsiooni loogiline raskus) ning arvutamise aeg (kahendteisendusoperatsioonide arv). Kriteeriumid on välja töötatud klassikaliste ja kiirete algoritmide tarvis. See annab võimaluse meetodeid kõrvutavalt analüüsida.

Viktoria CHERNYSHEVA

PROBLEMS OF CODING AND DECODING PROGRAM REALIZATION ESTIMATION

The present paper deals with the estimation of coding/decoding program realization. Complexity characteristics are determined for this purpose, such as: length of algorithm (logical complexity of function) and time of calculation (binary operations to realize the method). It has been derived for classic and high-speed algorithms. This can be used for comparative analysis of coding/decoding methods.