EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 32. KÖIDE
FÜÜSIKA * MATEMAATIKA. 1983, NR. 3

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 32
ФИЗИКА * МАТЕМАТИКА. 1983, № 3

УДК 62—519.007

В. ЧЕРНЫШЕВА

ОЦЕНКА СЛОЖНОСТИ МЕТОДОВ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ

(Представил Б. Тамм)

Современная теория кодирования обладает широким арсеналом методов повышения помехоустойчивости передачи информации. Цель корректирующего кодирования при этом заключается в согласовании высоких требований к достоверности передачи данных с низким качеством реальных каналов, в которых вероятность ошибки при передаче одного бита информации на 3—4 порядка больше допустимой. Многообразие корректирующих кодов, различие и сложность их математического аппарата в свою очередь затрудняют выбор и реализацию способа кодирования и декодирования.

Определим степень сложности реализации кодера/декодера с помощью следующих характеристик:

$W(K)$ — число запоминаемых двоичных символов при кодировании ($W(D)$ — при декодировании);

$W_o(K)$ — число преобразующих операций, необходимых при кодировании ($W_o(D)$ — при декодировании).

Утверждение 1. *При использовании линейного кодирования для одной кодовой последовательности*

$$W(K)=kn, \qquad W_o(K)=2kn.$$

Для преобразования информационной последовательности $u=\{u_1, \ldots, u_k\}$ в соответствующее кодовое слово $x=\{x_1, \ldots, x_n\}$ в кодере необходимо хранить порождающую матрицу G размерности $k \times n$ двоичных символов, где k — число информационных символов, n — длина кодового слова. Кодовые слова есть все возможные линейные комбинации строк матрицы G, $x=uG$. По правилам умножения матриц для каждой позиции кодового слова требуется не более k умножений и k сложений. Для n позиций кодового слова

$$W_o(K)=2kn.$$

Наиболее распространенным для линейного кода является синдромное декодирование.

Утверждение 2. *Если линейный код используется только для обнаружения ошибок и применяется синдромное декодирование, то вычисляемые характеристики принимают значения*

$$W(D)=rn, \qquad W_o(D)=2rn.$$

Пусть x^* — полученное кодовое слово. Для любого x^* вычисляется вектор $s=x^*H^{\tau}$ с $n-k=r$ компонентами, называемый синдромом. H — проверочная матрица размерности $r \times n$ двоичных символов. Поскольку некоторый вектор является кодовым словом тогда и только тогда, когда его синдром равен нулю, то ненулевой синдром указывает на наличие ошибок. В памяти необходимо хранить матрицу H, а по пра-

вилам умножения матриц требуется n умножений и n сложений; таким образом, $W_o(D) = 2rn$, $W(D) = rn$.

Утверждение 3. *Для линейного кода, обнаруживающего и исправляющего ошибки, при синдромном декодировании*

$$W(D) = n(r+2^n),$$

$$W_o(D) \leqslant n(2r+2^n-2^r+1).$$

Для линейных кодов, обнаруживающих и исправляющих ошибки, алгоритм декодирования расширяется. При ненулевом синдроме полученный кодовый вектор сравнивается со словами в таблице стандартного расположения, содержащей $2^r \times 2^k$ кодовых слов. После совпадения с некоторым кодовым словом в одном из 2^r смежных классов искомое слово получается в результате сложения полученного кодового слова с лидером соответствующего смежного класса. Число операций при этом складывается из числа операций для вычисления синдрома — $2rn$ — и максимального числа сравнений, что требует $2^n - 2^r$ сложений по модулю 2 с полученным кодовым вектором и сложения ошибочного кодового слова с найденным лидером. Таким образом, $W_o(D) \leqslant 2rn + n(2^n - 2^r) + n \leqslant n(2r + 2^n - 2^r + 1)$. В памяти помимо матрицы H необходимо хранить таблицу стандартного расположения, т. е. $W(D) = rn + 2_n{}^n = n(r + 2^n)$.

Следствие 3.1. *Сложность синдромного декодирования, рассмотренного в утверждении 3, можно уменьшить, получив следующие значения характеристик:*

$$W(D) = rn + 2^r(n+r),$$

$$W_o(D) = r2^r + n(2r+1).$$

Процесс декодирования может быть значительно упрощен за счет использования в качестве таблицы стандартного расположения таблицы, в которой приводятся 2^r лидеров смежных классов и соответствующие им синдромы порядка r. По вычисленному синдрому непосредственно находится вектор ошибки.

Следствие 3.2. *Для кода Хэмминга, исправляющего одну ошибку,* $W_o(D) = n(3r + 2)$.

Для кода Хэмминга, исправляющего одну ошибку, ненулевой синдром совпадает с тем столбцом матрицы H, порядковый номер которого равен порядковому номеру ошибки в кодовом векторе. Для вычисления синдрома согласно утверждению 2 требуется $2rn$ операций, максимальное число сравнений синдрома со столбцами матрицы H равно nr. Для нахождения вектора ошибки и последующего его сложения с полученным кодовым вектором требуется $2n$ операций

$$W_o(D) = 2nr + nr + 2n = n(3r+2).$$

Утверждение 4. *Для циклического кода, обнаруживающего ошибки,* $W(K) = W(D) = r+1$; $W_o(K) = W_o(D) = n(k+1)+k$.

Для систематического циклического кода кодирование сводится к делению на генераторный полином $g(x)$ длиной $r + 1$, $W(K) = W(D) = r + 1$. Этот процесс требует k сдвигов и не более $k + 1$ сложений с кодовым словом, т. е. $W_o(K) = n(k + 1) + k$. Процесс декодирования для кода, обнаруживающего ошибки, аналогичен процессу кодирования, и поэтому $W_o(D) = W_o(K)$.

Исправление ошибок может происходить несколькими путями.

Утверждение 5. *Для циклического кода, декодирование которого происходит по алгоритму Прейнджа,*

$$W_o(D) \leqslant n(k+1)^2 + 16k; \qquad W(D) = r + 17.$$

Согласно алгоритму Прейнджа [1], определение места ошибки происходит следующим образом. Вычисляется вес w остатка от деления кодового слова на генераторный полином $g(x)$. Величина w сравнивается с величиной $w_0 = \left]\frac{d-1}{2}\right[$, где d — кодовое расстояние. Если $w \leqslant w_0$, то искомый кодовый вектор является результатом сложения полученной комбинации и вычисленного остатка. Иначе, после циклического сдвига кодового вектора процесс декодирования повторяется до выполнения данного условия. Искомое кодовое слово получается в результате сложения сдвинутой комбинации с остатком и в обратном циклическом сдвиге на произведенное число сдвигов. На каждом этапе производится $n(k+2)+16$ операций, число сдвигов не может превышать числа информационных символов k.

$$W_o(D) = k[n(k+2)+16]+n = n(k+1)^2+16k.$$

Утверждение 6. *Для циклического кода, исправляющего ошибки кратности t, декодирование которого происходит по алгоритму Меггита,* $W_o(D) \leqslant k(n+2r+4)$.

Согласно алгоритму Меггита [2], вычисляется остаток от деления полученного слова на $g(x)$, названный синдромом. Между синдромом $s(x)$ и комбинацией ошибок существует взаимно однозначное соответствие, что позволяет после циклического сдвига синдрома получить новый синдром, соответствующий синдрому сдвинутой комбинации ошибок. Обнаружение местоположения ошибок возможно, когда после i сдвигов они окажутся в высших разрядах, что позволяет исправить $i, \ldots, i-t-1$ ошибки. Число сдвигов и сложений определяется числом информационных символов. Поэтому $W_o(D) = n(k+1) + k+k+(k-1)(r+1)+kr = k(n+2r+4)$.

Утверждение 7. *Для древовидного кодирования*

$$W(K) = (k_0+n_0)2^{k_0}; \qquad W_o(K) < k_e 2^{k_0}+n_e.$$

Для древовидных кодов объем памяти связан с кодовым ограничением $m_e n_0$, определяющим количество хранимых на каждом этапе ветвей кода, и длиной отрезка k_0, на которые разбивается информационная последовательность. Из каждого узла древовидного кода выходит 2^{k_0} ветвей. Переход на ту или иную ветвь происходит в соответствии с одним из 2^{k_0} правил, т. е. в памяти необходимо хранить 2^{k_0} ветвей длиной n_0 двоичных символов и 2^{k_0} правил перехода длиной k_0 двоичных символов, тогда $W(K) = (k_0+n_0)2^{k_0}$. Для получения кодовой последовательности, ограниченной длиной $m_e n_0$, необходимо произвести $m_e 2^{k_0}$ сравнений и $m_e n_0$ сложений

$$W_o(K) = m_e k_0 2^{k_0}+m_e n_0 = k_e 2^{k_0}+n_e.$$

Частным случаем древовидных кодов являются сверточные коды, обладающие наиболее простой реализацией по сравнению с другими древовидными кодами.

Утверждение 8. *Для сверточных кодов*

$$W(K) = k_e n_0^2; \qquad W_o(K) = k_e n_0^2 (n_e+1).$$

Для сверточного (mn_0, mk_0)-кода порождающая матрица имеет вид

$$G = \begin{bmatrix} G_0 & G_1 & \ldots & G_{m-1} \\ 0 & G_0 & \ldots & G_{m-2} \\ \vdots & \ddots & \ddots & \ddots \\ & & & G_0 \end{bmatrix},$$

где G_j — матрица размерности $k_0 \times n_0$ двоичных символов. Для вычислений достаточно хранить m матриц G_j, $W(K) = m_e n_0 k_0 n_0 = k_e n_0^2$. Число операций для получения первых n_0 символов равно $2k_0n_0$ (см. утверждение 1), для последующих — $4k_0n_0$, $6k_0n_0 \ldots 2m_en_0k_0n_0$ соответственно. Суммируя, получим искомую величину $W_o(K)$.

Утверждение 9. *Для древовидных кодов, использующих при декодировании алгоритм Витерби,*

$$W(D) \leqslant n_0 2^{m_e n_0 R}(m_e + mR), \quad W_o(D) \leqslant 2^{k_e+1}(n_e + (m_e - 1)(m - m_e)).$$

Применяя декодер Витерби [2], для принятой последовательности $r = r_0, r_1, \ldots, r_{2^{k_e}}$ вычисляются значения расстояний $d(c_i, r) = \sum_{j=0}^{m_e-1} d(c_{ij}, r_j)$, $i = \overline{0, 2^{k_e} - 1}$, где r_j — двоичный набор длины n_0, $\{c_{ij}\}$, $i = \overline{0, 2^{k_e} - 1}$, кодовые слова длины n_e. Затем сравниваются расстояния, соответствующие 2^{k_0} путям, для которых последние $m_e - 1$ информационных блоков равны. Путь с наименьшим расстоянием называется выжившим. На втором этапе вычисляются расстояния между r и 2^{k_e} наборами $n_0(m_e + 1)$, образованными удлинением на одну ветвь каждого выжившего пути, сравниваются и определяются новые выжившие пути. Действия выполняются до тех пор пока первые символы не совпадут. Декодер должен хранить $n_e 2^{k_e}$ двоичных символов кодовых слов, $k_e 2^{k_e}$ двоичных символов информационных последовательностей, $2^{k_e+1} \log_2 n_e$ двоичных символов расстояний $d(c_i, r)$ и матрицу G. Число этапов определяется ограничением при декодировании mn_0, которое обычно в 3—4 раза превышает длину кодового ограничения, и равно $m - m_e + 1$.

Для определения расстояния на первом этапе требуется $2n_e$ операций для каждого из 2^{k_e} кодовых слов и по $2(m_e - 1)$ операций на последующих $m - m_e$ этапах. Таким образом,

$$W_o(D) = 2n_e 2^{k_e} + 2(m_e - 1)2^{k_e}(m - m_e) = 2^{k_e+1}(n_e + (m_e - 1)(m - m_e)).$$

Оптимальным методом, минимизирующим среднюю вероятность ошибки, является декодирование по максимуму правдоподобия. Но оно неудовлетворительно, так как в декодере необходимо хранить список всех слов кода и последовательно сравнивать последовательность со всеми словами списка. При использовании других методов декодирования увеличивается составляющая вероятности ошибки, которая обуславливается помехами в канале, но уменьшается составляющая, которая зависит от ошибок в самом декодирующем устройстве. Поэтому особый интерес представляет один из неоптимальных алгебраических методов декодирования — мажоритарное декодирование.

Утверждение 10. *Для линейного кода, использующего мажоритарное декодирование с системой разделенных проверок,*

$$W(D) = Jkn, \quad W_o(D) \leqslant 2kJ(n - 1).$$

Системой разделенных проверок называется такое множество контрольных проверок, в котором некоторый символ a_i входит в каждую контрольную проверку, а любой другой символ a_j, $j \neq i$ входит не более чем в одну проверку. Для правильного декодирования одного символа, если при передаче исказились $e \leqslant t$ символов кодового слова, достаточно, чтобы система разделенных проверок для одного символа содержала не менее $2t + 1$ контрольных соотношений. Для k символов сообщения линейного кода необходимо k систем разделенных проверок.

На каждом этапе декодирования на основе системы разделенных проверок по «большинству» определяется декодированный символ, что

требует $2J(n-1)$ операций. Для всех информационных символов $W_o(D) \leqslant 2kJ(n-1)$, где J — число нетривиальных проверок в системе разделенных проверок.

Число методов кодирования и декодирования растет. Эффективное кодирование и декодирование достигается обеспечением малой вероятности ошибки, малой вероятности ошибочного декодирования, обеспечением номинального объема памяти и числа преобразующих операций.

ЛИТЕРАТУРА

1. *Колесник В. Д., Мирончиков Е. Т.* Декодирование циклических кодов. М., «Связь», 1968.
2. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М., «Мир», 1976.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
28/I 1983

V. TSERNOSEVA

KODEERIMIS- JA DEKODEERIMISMEETODITE KEERUKUSE HINDAMINE

Suurest hulgast kodeerimis- ja dekodeerimismeetoditest efektiivseima valimise teeb raskeks matemaatilise aparatuuri erinevus ja keerukus. Käesolevas artiklis on kooderi/dekooderi realisatsioone iseloomustatud mällu salvestatavate kahendsümbolite arvu ning kahendteisendusoperatsioonide arvu järgi.

V. CHERNYSHEVA

CODING AND DECODING COMPLEXITY ESTIMATION

The present paper deals with the problem of the choice and realization of coding and decoding methods for the improvement of data transmission reliability. Coder/decoder complexity characteristics are determined for this purpose, as:

$W(K)$ — binary symbols to retain for coding ($W(D)$ — for decoding method);

$W_o(K)$ — binary operations to realize for coding ($W_o(D)$ — for decoding method).

Here, some of the advanced coding/decoding methods are considered. $W(K)$, $W_o(K)$ have been derived for linear, Hamming, cyclic, woodlike (recurrent), and convolutional codes. $W(D)$, $W_o(D)$ have been derived for syndrom decoding, for error detecting and error correcting codes, for decoding by Prange's algorithm, by Meggit's algorithm, by Viterbi's algorithm, and for majoritary logical decoding.