Распараллеливание вычислений при декодировании недвоичных кодов с малой плотностью проверок*

И. В. Жилин
ИППИ РАН
zhilin@iitp.ru

Ф. И. Иванов
ИППИ РАН
fii@iitp.ru

В. В. Зяблов
ИППИ РАН
zyablov@iitp.ru

Аннотация

Предложена модификация алгоритма декодирования q-ary Sum Product Algorithm (Q-SPA) для недвоичных кодов с малой плотностью проверок, основанных на матрицах перестановок. Вычисления, используемые при декодировании такого класса кодов, могут быть естественным образом распараллелены, и таким образом представленный в работе алгоритм имеет векторную реализацию, работая не с отдельными символами над полем $GF(q)$, а с векторами, заданными над этим полем.

1. Введение

Двоичные коды с малой плотностью проверок (МПП-коды) были предложены в 1960 году Галлагером [1]. Данные линейные блочные коды задаются с помощью проверочной матрицы $\mathbf{H}$, характеризуемой относительно малым числом единиц в строках и столбцах. Часто проверочную матрицу $\mathbf{H}$ МПП-кода удобно представлять в виде графа Таннера [2]. Так же в [1] Галлагер дал краткое описание МПП-кодов, заданных над произвольным полем $GF(q)$, а так же представил набросок алгоритма их декодирования по апостериорным вероятностям на выходе канала (алгоритм "распространения доверия").

Непосредственно после своего открытия МПП-коды не получили широкого распространения ввиду достаточно сложной реализации и были практически забыты на протяжении последующих 30 лет. Тем не менее в конце 1990-х годов они были переоткрыты [3,4], и с тех пор интерес к данному классу кодов все более и более усиливается. Однако, в подавляющем большинстве работ, посвященных МПП-кодам, рассматриваются двоичные коды.

Среди публикаций, в которых рассматриваются недвоичные МПП-коды, следует особо отметить работу [4], в которой подробно описан ансамбль МПП-кодов над полем $GF(q)$, а так же обобщенный алгоритм их декодирования $q-ary\,Sum\,Product\,Algorithm\,(Q-SPA)$. В работе [5] представлено улучшение данного алгоритма, основанное на использовании быстрого преобразования Фурье, что позволяет существенно снизить вычислительную сложность декодера.

Поиск способов эффективного хранения проверочных матриц недвоичных МПП-кодов стал причиной разработки новых методов их построения. В частности, для генерации проверочных матриц таких кодов стали использоваться матрицы перестановок [6,7,8].

В данной работе рассмотрена модификация алгоритма декодирования $Q-SPA$ для случая, когда проверочная матрица $\mathbf{H}$ недвоичного кода с малой плотностью проверок состоит из произвольных матриц перестановок, умноженных на произвольные диагональные матрицы (с элементами из мультипликативной группы поля $GF^\star(q)$. Основное преимущество данного алгоритма заключается в том, что он имеет параллельную реализацию, работая не с отдельными символами над полем $GF(q)$, а с векторами, заданными над этим полем.

2. Недвоичные МПП-коды, основанные на матрицах перестановок

В данном разделе мы дадим наиболее общее описание недвоичных кодов с малой плотностью проверок, основанных на матрицах перестановок.

Определение 2.1 *Пусть $l, n_0, m \in \mathbb{N}$, причем $2 < l < n_0$, $ln_0 > m!$. Рассмотрим циклическую мультипликативную группу $GF^\star(q) = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$ поля $GF(q)$, $q = p^t$, $t \geq 1$, $p \geq 2$ – простое, α – примитивный элемент поля, а так же симметрическую группу $\mathscr{P}_m$ матриц перестановок размерности m, $|\mathscr{P}_m| = m!$.*

В симметрической группе выберем ln_0 случайных матриц $\{\mathbf{P}_{ji}\} \in \mathscr{P}_m$, $j = 1..l, i = 1..n_0$. Потребуем также, что если $P_{ji} = P_{ks}$, то $j = k$, $i = s$. Ясно, что

*Работа выполнена при частичной поддержке РФФИ (грант № 12-07-31035 мол_а).

такие условия выбора матриц перестановок $\mathbf{P}_{ji}$ *соответствуют урновой модели без возвращений.*

Составим ln_0 *случайных* m*-элементных векторов (возможно, с повторениями элементов внутри набора), состоящих из элементов группы* $GF^\star(q)$*:* $S_{ji} = (\beta_{ji}^{(1)}, \beta_{ji}^{(2)}, \ldots, \beta_{ji}^{(m)})$*,* $j = 1..l$*,* $i = 1..n_0$*.*

Каждую из ln_0 $m \times m$ *матриц* $\mathbf{P}_{ji}$ *домножим на диагональную матрицу следующего вида:*

$$\mathbf{I}_{\mathbf{S}_{ji}} = diag(S_{ji}) = \begin{pmatrix} \beta_{ji}^{(1)} & 0 & \ldots & 0 \\ 0 & \beta_{ji}^{(2)} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \beta_{ji}^{(m)} \end{pmatrix}.$$

Полученные ln_0 *матриц вида:*

$$\mathbf{R}_{ji} = \mathbf{P}_{ji} \cdot \mathbf{I}_{\mathbf{S}_{ji}}$$

сделаем элементами блочной матрицы $\mathbf{H}$*:*

$$\mathbf{H} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \ldots & \mathbf{R}_{1n_0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \ldots & \mathbf{R}_{2n_0} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbf{R}_{l1} & \mathbf{R}_{l2} & \ldots & \mathbf{R}_{ln_0} \end{pmatrix}.$$

Указанный выше способ построения матрицы $\mathbf{H}$ *гарантирует, что все матрицы в каждой строке и каждом столбце будут различны. Так как* $\mathbf{R}_{ij}$ *- квадратная* $m \times m$ *матрица, то размерность* $\mathbf{H}$ *-* $ml \times mn_0$*.*

$\mathbf{H}$ *определяет ансамбль регулярных* q*-ичных* (l, n_0)*-кодов с малой плотностью проверок длины* $n = mn_0$*, который мы обозначим* $\mathscr{E}_P(l, n_0, m, q)$*. Элементы ансамбля* $\mathscr{E}_P(l, n_0, m, q)$ *получаются путем выбора без возвращений матриц перестановок* $\mathbf{P}_{ji} \in \mathscr{P}_m$*,* $j = 1..l$*,* $i = 1..n_0$*, а так же* ln_0 m*-элементных наборов* $S_{11}, \ldots, S_{ln_0}$*.*

Произвольный код $\mathscr{C} \in \mathscr{E}_P(l, n_0, m, q)$ *назовем* q*-ичным кодом с малой плотностью проверок, основанным на матрицах перестановок.*

Одним из наиболее распространенных на практике и простых по структуре подансамблей ансамбля $\mathscr{E}_P(l, n_0, m, q)$ является ансамбль недвоичных "квазициклических" МПП-кодов, который мы будем обозначать $\mathscr{E}_Q(l, n_0, m, q)$. Данный ансамбль получается, если в качестве матриц $\mathbf{P}_{ji}$ выбираются элементы из подгруппы $\mathscr{H} \in \mathscr{P}_m$. $\mathscr{H}$ состоит из всех циклических сдвигов столбцов единичной $m \times m$ матрицы $\mathbf{I}$. Очевидно, что $|\mathscr{H}| = m$.

Произвольный код $\mathscr{C} \in \mathscr{E}_Q(l, n_0, m, q)$ назовем недвоичным "квазициклическим" МПП-кодом.

Как уже было отмечено, ансамбль $\mathscr{E}_Q(l, n_0, m, q)$ является подансамблем ансамбля $\mathscr{E}_P(l, n_0, m, q)$. В то же время, поскольку каждая из матриц $\mathbf{P}_{ji}$ недвоичного "квазициклического" МПП-кода полностью определяется одним целым числом – величиной циклического сдвига $0 \le r_{ji} \le m - 1$, то для хранения всей проверочной матрицы $\mathbf{H}$ нам достаточно $ln_0(m \log_2 q + \log_2 m)$ бит. В то же время легко заметить, что для хранения полной матрицы из ансамбля $\mathscr{E}_P(l, n_0, m, q)$ потребовалось бы $mln_0(\log_2 q + \log_2 m)$ бит. Таким образом, обеспечивается почти m -кратная экономия памяти по сравнению с произвольным кодом из ансамбля $\mathscr{E}_P(l, n_0, m, q)$.

В то же время, для хранения проверочной матрицы q-ичного МПП-кода Галлагера [1], требуется $mln_0(\log_2 q + \log_2 n)$ бит, таким образом, использование кода из ансамбля $\mathscr{E}_Q(l, n_0, m, q)$ так же приводит примерно к m-кратной экономии памяти.

3. Вычисление синдрома для недвоичного МПП-кода, основанного на матрицах перестановок

Пусть

$$\mathbf{H} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \ldots & \mathbf{R}_{1n_0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \ldots & \mathbf{R}_{2n_0} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbf{R}_{l1} & \mathbf{R}_{l2} & \ldots & \mathbf{R}_{ln_0} \end{pmatrix}$$

проверочная матрица регулярного q-ичного (l, n_0)-кода с малой плотностью проверок, причем размер матрицы $\mathbf{R}_{ji}$ равен $m \times m$. Матрицу $\mathbf{H}$ можно представить в виде двух матриц

$$\mathbf{H}_\pi = \begin{pmatrix} \pi_{11} & \pi_{12} & \ldots & \pi_{1n_0} \\ \pi_{21} & \pi_{22} & \ldots & \pi_{2n_0} \\ \ldots & \ldots & \ldots & \ldots \\ \pi_{l1} & \pi_{l2} & \ldots & \pi_{ln_0} \end{pmatrix},$$

$$\mathbf{H}_S = \begin{pmatrix} S_{11} & S_{12} & \ldots & S_{1n_0} \\ S_{21} & S_{22} & \ldots & S_{2n_0} \\ \ldots & \ldots & \ldots & \ldots \\ S_{l1} & S_{l2} & \ldots & S_{ln_0} \end{pmatrix},$$

где π_{ji} – перестановка, соответствующая матрице $\mathbf{P}_{ji}$, а вектор S_{ji} аналогичен рассмотренному в определении 2.1.

Поскольку длина МПП-кода с проверочной матрицей $\mathbf{H}$ равна $n = mn_0$, то кодовое слово $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, $c_i \in GF(q)$, можно представить в следующем виде:

$$\mathbf{c} = (\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_{n_0}),$$

где $\bar{\mathbf{c}}_i$ – q-ичный вектор длины m. Напомним, что синдром $\mathbf{S}$ для принятого слова $\mathbf{y}$ вычисляется по формуле $\mathbf{S} = \mathbf{H}\mathbf{y}^\mathbf{T}$, причем $\mathbf{S} = \mathbf{0}$ тогда и только тогда, когда $\mathbf{y}$ является кодовым словом. Последнее соотношение может быть записано в виде однородной системы из l уравнений:

$$\begin{cases} \pi_{11}(S_{11} \cdot \bar{\mathbf{c}}_1) + \pi_{12}(S_{12} \cdot \bar{\mathbf{c}}_2) + \ldots + \pi_{1n_0}(S_{1n_0} \cdot \bar{\mathbf{c}}_{n_0}) = \mathbf{0} \\ \pi_{21}(S_{21} \cdot \bar{\mathbf{c}}_1) + \pi_{22}(S_{22} \cdot \bar{\mathbf{c}}_2) + \ldots + \pi_{2n_0}(S_{2n_0} \cdot \bar{\mathbf{c}}_{n_0}) = \mathbf{0} \\ \ldots \\ \pi_{l1}(S_{l1} \cdot \bar{\mathbf{c}}_1) + \pi_{l2}(S_{l2} \cdot \bar{\mathbf{c}}_2) + \ldots + \pi_{ln_0}(S_{ln_0} \cdot \bar{\mathbf{c}}_{n_0}) = \mathbf{0} \end{cases}$$

Где умножения векторов внутри перестановок π_{ij} выполняются поэлементно и по правилам группы $GF^{\star}(q)$. Сложения происходят по правилам поля $GF(q)$. Таким образом, доказано следующее

Утверждение 3.1 *Вектор* $\mathbf{y} = (\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_{n_0})$, *где* $\bar{\mathbf{y}}_i$ – *q-ичный вектор длины* m, *является кодовым словом кода с малой плотностью проверок длины* $n = mn_0$, *задаваемого проверочной матрицей* $\mathbf{H}$, *тогда и только тогда, когда выполняется* l *соотношений*

$$\sum_{i=1}^{n_0} \pi_{ji}(S_{ji} \cdot \bar{\mathbf{c}}_i) = \mathbf{0},\ j = 1..l.$$

Как следует из утверждения, для q-го МПП-кода, основанного на матрицах перестановок, вычисление синдрома ошибки имеет векторный характер: в вычислениях используются не отдельные q-ичные символы принятого слова, а блоки длины m.

4. Декодирование недвоичных МПП-кодов, основанных на матрицах перестановок

В данной работе предложена модификация алгоритма $q-ary\,Sum Product Algorithm$ для недвоичных кодов с малой плотностью проверок, основанных на матрицах перестановок. Основная идея предложенной модификации заключается в одновременной обработке m символов принятого слова (т.е. алгоритм работает с векторами длины m), в то время как классический алгоритм $Q-SPA$ не предусматривает такую возможность. Векторный характер декодирования принятого слова, как будет показано, позволяет распараллелить декодер в m раз, что существенно отразится на скорости обработки данных.

Отметим, что схожий алгоритм декодирования для случая двоичных МПП-кодов, основанных на матрицах перестановок, приведен в работе [9].

Как и при декодировании случайного q-ичного МПП-кода, на вход алгоритму передается оценка вероятностного распределения символов, полученная из канала [5]. Данная оценка представляет из себя матрицу $\mathbf{L}$ размерности $q \times n$, где i-му столбцу $\mathbf{L}$ соответствует вектор-столбец правдоподобий

$$\mathbf{L}_i = (p(x_i = 0), p(x_i = 1), \ldots, p(x_i = q-1))^T$$

где, что $p(x_i = \gamma)$ обозначает вероятность того, что i-й символ равен $\gamma \in GF(q)$. Будем называть такие вектор-столбцы мягкими значениями q-ичных символов. Далее по тексту под векторами будет позразумевать вектор-строки. Тогда о матрице $\mathbf{L}$ мы можем говорить как о вектор-строке мягких значений входных символов.

В данном алгоритме декодирования будет использоваться два типа операций: умножение мягкого q-ичного значения на ненулевое жёсткое q-ичное значение (на элемент группы $GF^{\star}(q)$), и сложение мягких q-ичных значений. Стоит отметить, что результатом операции над мягкими q-ичными значениями $\mathbf{L}_i = \mathbf{L'}_i \diamond \mathbf{L''}_i$ является такое мягкое q-ичное значения, вероятности $p(x_i = \gamma)$ которого являются вероятностями того, что результат операции будет иметь значение γ при заданных входных распределениях вероятностей $\mathbf{L'}_i$ и $\mathbf{L''}_i$.

Умножение мягкого q-ичного значения $\mathbf{L'}_i$ на жёсткое q-ичное значение из $\alpha^k \in GF^{\star}(q)$ по сути является перестановкой столбцов $\mathbf{L'}_i$. Будем его обозначать как $\mathbf{L}_i = \mathbf{L'}_i \odot \alpha^k$.

Сложение двух мягких q-ичных значений будем обозначать $\mathbf{L}_i = \mathbf{L'}_i \oplus \mathbf{L''}_i$. Оно является свёрткой векторов распределениях вероятностей $\mathbf{L'}_i$ и $\mathbf{L''}_i$ [5].

Под произведениями и суммами векторов в описании алгоритма подразумеваются их поэлементные произведения и суммы соответственно.

Так как $n = mn_0$, то для матрицу $\mathbf{L}$ можно представить в следующем виде:

$$\mathbf{L} = \left[\bar{\mathbf{L}}_1 \bar{\mathbf{L}}_2 \ldots \bar{\mathbf{L}}_{n_0}\right],$$

где

$$\bar{\mathbf{L}}_i = \begin{pmatrix} p(x_{(i-1)m+1} = 0) & \ldots & p(x_{im} = 0) \\ p(x_{(i-1)m+1} = 1) & \ldots & p(x_{im} = 1) \\ \ldots & \ldots & \ldots \\ p(x_{(i-1)m+1} = q-1) & \ldots & p(x_{im} = q-1) \end{pmatrix}.$$

Введем множество $I(j)$ – набор переменных $\{v_1^{(j)}, v_2^{(j)}, \ldots, v_{n_0}^{(j)},\}$ участвующих в j-й проверке, и множество $J(i)$ – набор проверок $\{c_1^{(i)}, c_2^{(i)}, \ldots, c_l^{(i)},\}$ в которые входит i-я переменная. Рассмотрим произвольную матрицу $\bar{\mathbf{L}}_i$. Так как размерность $\bar{\mathbf{L}}_i$ равна $q \times m$, а матрицы $\mathbf{R}_{1i}, \mathbf{R}_{2i}, \ldots, \mathbf{R}_{li}$ – $m \times m$ матрицы содержащие ровно один ненулевой элемент в каждой строке и каждом столбце, то $\bar{\mathbf{L}}_i$ участвует в ml различных проверках. Таким образом, $|J(\bar{\mathbf{L}}_i)| = ml$. Полученное равенство позволяет нам сделать вывод о том, что элементы $\bar{\mathbf{L}}_i$ участвуют во всех проверках. Таким образом, при декодировании с использованием алгоритма $Q-SPA$ нам не требуется искать $J(\bar{\mathbf{L}}_i)$ для каждого вектора $\bar{\mathbf{L}}_i$. Проводя аналогичные рассуждения, можно показать, что в m проверках участвуют mn_0 переменных, поэтому вычисление $I(j)$ для каждой j-й проверки также не требуется.

Введем необходимые обозначения:

$\mathbf{L} = \left[\bar{\mathbf{L}}_1 \bar{\mathbf{L}}_2 \ldots \bar{\mathbf{L}}_{n_0}\right]$ — принятый из канала вектор столбцов правдоподобий;

$\mathbf{L'} = \left[\bar{\mathbf{L}'}_1 \bar{\mathbf{L}'}_2 \ldots \bar{\mathbf{L}'}_{n_0}\right]$ — вычисленный вектор столбцов правдоподобий;

α_{ji} – строка сообщений от переменных $\bar{\mathbf{L}}_i$ к j-й группе из m проверок;

γ_{ji} – строка сообщений от j-й группы из m проверок к $\bar{\mathbf{L}}_i$;

Изложенный ниже алгоритм декодирования применим для МПП-кодов, основанных на матрицах перестановок и работает с проверочной матрицей, представленной в форме:

$$\mathbf{H}_\pi = \begin{pmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n_0} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n_0} \\ \dots & \dots & \dots & \dots \\ \pi_{l1} & \pi_{l2} & \dots & \pi_{ln_0} \end{pmatrix},$$

$$\mathbf{H}_S = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1n_0} \\ S_{21} & S_{22} & \dots & S_{2n_0} \\ \dots & \dots & \dots & \dots \\ S_{l1} & S_{l2} & \dots & S_{ln_0} \end{pmatrix},$$

где π_{ji} – перестановка, соответствующая матрице $\mathbf{P}_{ji}$, а вектор S_{ji} аналогичен рассмотренному в определении 2.1.

Декодирование включает в себя следующие шаги:

1. *Начальная проверка:* по принятому из канала вектору $\mathbf{L} = \left[\bar{\mathbf{L}}_1 \bar{\mathbf{L}}_2 \dots \bar{\mathbf{L}}_{n_0}\right]$ строится жёсткое решение $\mathbf{x}$, вычисляется синдром $\mathbf{Hx^T}$ согласно алгоритму, описанному в разделе 3. Если синдром равен нулевому вектору, то декодирование прекращается и $\mathbf{x}$ являетс результатом выполнения алгоритма, иначе переходим к шагу 2.

2. *Инициализация:* Строим матрицу $\mathbf{A}$ по правилу:

$$\mathbf{A} = \begin{pmatrix} \bar{\alpha}_{11} & \bar{\alpha}_{12} & \dots & \bar{\alpha}_{1n_0} \\ \bar{\alpha}_{21} & \bar{\alpha}_{22} & \dots & \bar{\alpha}_{2n_0} \\ \dots & \dots & \dots & \dots \\ \bar{\alpha}_{l1} & \bar{\alpha}_{l2} & \dots & \bar{\alpha}_{ln_0} \end{pmatrix},$$

где

$$\bar{\alpha}_{ji} = \pi_{ji}(S_{ji} \odot \bar{\mathbf{L}}_i)$$

3. *Горизонтальный шаг:* Строим матрицу Γ по правилу:

$$\Gamma = \begin{pmatrix} \bar{\gamma}_{11} & \bar{\gamma}_{12} & \dots & \bar{\gamma}_{1n_0} \\ \bar{\gamma}_{21} & \bar{\gamma}_{22} & \dots & \bar{\gamma}_{2n_0} \\ \dots & \dots & \dots & \dots \\ \bar{\gamma}_{l1} & \bar{\gamma}_{l2} & \dots & \bar{\gamma}_{ln_0} \end{pmatrix},$$

где

$$\bar{\gamma}_{ji} = \bigoplus_{t=1, t\neq i}^{n_0} \bar{\alpha}_{jt}$$

4. *Вертикальный шаг:* Вычисление вектора апостериорных вероятностей и сообщений от переменных к проверкам:

$$\bar{\mathbf{L}'}_i = \bar{\mathbf{L}}_i \odot \left[\bigodot_{j=1}^{l} \pi_{ji}^{-1}(\bar{\gamma}_{ji}) \odot S_{ji}^{-1}\right]$$

$$\bar{\alpha}_{ji} = \bar{\mathbf{L}}_i \odot \left[\bigodot_{t=1, t\neq j}^{l} \pi_{ti}^{-1}(\bar{\gamma}_{ti}) \odot S_{ti}^{-1}\right]$$

5. *Проверка синдрома:* По вычисленным $\bar{\mathbf{L}'}_i$ строится жёсткое решение $\mathbf{x}$ и вычисляется синдром $\mathbf{S} = \mathbf{Hx^T}$. Если $\mathbf{S} = \mathbf{0}$, то декодирование прекращается и $\mathbf{x}$ считается результатом выполнения алгоритма, иначе переходим к шагу 3.

Шаги 3–5 выполняются ограниченное число раз. Если достигнуто максимальное число итераций, то алгоритм прерывается и блок считается принятым с ошибкой.

Описанный выше алгоритм оперирует только с векторами длины m, не обращаясь явно к отдельным символам. Таким образом, процесс декодирования можно осуществлять параллельно для m символов.

5. Заключение

Предложен векторный алгоритм декодирования недвоичных МПП-кодов, основанных на матрицах перестановок. Данный алгоритм декодирования ориентирован на векторные вычислители, такие, как видеокарта, программируемая логическая интегральная схема (ПЛИС) и т. д. Для декодера осуществляется распараллеливание в m раз, где m достаточно велико. Данный подход позволяет существенно увеличить скорость декодирования. Поскольку к современным сигнально-кодовым конструкциям предъявляются достаточно жесткие требования по скорости обработки и передачи данных, то построенный декодер имеет практическую ценность.

Список литературы

[1] Р. Дж. Галлагер, *Коды с малой плотностью проверок*, Москва, Мир, 1966.

[2] M. A. Tanner, *Recursive Approach to Low Complexity Codes*, IEEE Trans. Inform. Theory, 1981, vol. 27, no. 5, pp. 533–547.

[3] M. Davey, D. J. C. MacKay, *Good Error-Correcting Codes Based on Very Sparse Matri ces*, IEEE Trans. Inform. Theory, 1999, vol. 45, no. 2, pp. 399–432.

[4] M. Davey, D. J. C. MacKay, *Low density parity check codes over GF(q)*, IEEE Commun. Lett., 1998, vol. 2, no. 6, pp. 165–167.

[5] D. Declercq, M. Fossorier, *Decoding Algorithms for Nonbinary LDPC Codes over GF(q)*, IEEE Trans. Commun., 2007, vol. 55, no. 4, pp. 633–643.

[6] S. Lin, S. Song, L. Lan, L. Zeng, Y. Y. Tai, *Constructions of Nonbinary Quasi-Cyclic LDPC Codes: A Finite Field Approach*, IEEE Trans. Commun., 2008, vol. 56, no. 4, pp. 545–554.

[7] S. Lin, S. Song, B. Zhou, *Algebraic constructions of non-binary quasi-cyclic LDPC codes: array masking and dispersion*, in Proc. 9th International Symposium on Communication Theory and Applications (ISCTA), 2007, Ambleside, Lake District, UK.

[8] R. A. Carrasco, M. Johnston, *Non-binary error control coding for wireless communication and data storage*, USA, Wiley, 2009.

[9] Ф. И. Иванов, И. В. Жилин, В. В. Зяблов, *Алгоритм декодирования кодов с малой плотностью проверок с большим распараллеливанием,* Информационно-управляющие системы, 2012, № 6, с. 49–55.