

УДК 621.391

В. Ш. Ле

Московский физико-технический институт (национальный исследовательский университет)

Оптимизация алгоритма декодирования min-sum для кодов с низкой плотностью проверок на четность

Рассмотрены итеративные алгоритмы декодирования кодов с низкой плотностью проверок на четность. Приведены различные модифицированные версии алгоритма min-sum. Построены графики зависимости битовой ошибки при передаче данных по двоичному каналу связи с аддитивным белым гауссовским шумом. Проведено сравнение сложности реализации различных алгоритмов декодирования.

Ключевые слова: LDPC-код, алгоритм «sum-product», алгоритм «min-sum», алгоритм «min-sum normalized», алгоритм «min-sum offset», комбинированный алгоритм «min-sum», коэффициент нормализации, коэффициент сдвига.

V. Sh. Le

Moscow Institute of Physics and Technology

Optimization of the min-sum decoding algorithm for low-density parity-check codes

In this paper iterative decoding algorithms of low-density parity check codes are considered. Various modified versions of the minsum algorithm are considered. Dependency graphs of the bit error on various coefficients when transmitting data over a binary communication channel with additive white Gaussian noise are constructed. Comparison of the complexity of the implementation of various decoding algorithms is studied.

Key words: LDPC-code, sum-product algorithm, min-sum algorithm, min-sum normalized algorithm, min-sum offset algorithm, combined min-sum algorithm, normalization factor, shift factor.

1. Введение

Среди многих существующих помехоустойчивых кодов коды с низкой плотностью проверок на четность (LDPC-коды) являются одними из самых мощных, эффективность которых приближается к пределу Шеннона [1, 2]. LDPC-коды впервые были представлены Робертом Галлагером [3] в 1963 г., однако из-за сложности реализации коды практически не нашли применения. С развитием компьютерных технологий и ростом требований к точности передачи информации LDPC-коды вновь привлекли к себе внимание исследователей. В последнее время эти коды применяются во многих современных стандартах, таких как DVB-S2, DVB-T2 и IEEE 802.3an [1, 4, 5].

Существует множество алгоритмов декодирования LDPC-кодов. Среди них самым известным является алгоритм «sum-product» [6], который обеспечивает высокую эффективность декодирования, но требует больших вычислительных затрат. В связи с этим существуют альтернативные методы, целью которых является снижение вычислительных затрат. Среди них одним из самых популярных является алгоритм «min-sum» [6, 7]. Алгоритм «min-sum» является аппроксимацией алгоритма «sum-product» с использованием простых

операций, таких как поиск минимума и сложение, вместо сложных функций гиперболического тангенса и арктангенса. В то же время из-за такого упрощения алгоритм «min-sum» обладает худшей помехоустойчивостью по сравнению с «sum-product» алгоритмом. Проигрыш по энергетическому выигрышу кодирования составляет при этом $0.2 \dots 0.5$ дБ [4]. В связи с чем в данной работе предложены некоторые модифицированные версии алгоритма «min-sum» для повышения его эффективности.

2. Алгоритмы декодирования LDPC-кодов

LDPC-код представляет собой линейный блочный код, характеризующийся разрежённой проверочной матрицей $H(N \times M)$, где M – количество проверочных строк; N – длина кода, т.е. количество столбцов в матрице; K – длина исходных данных. Если количество ненулевых элементов в каждой строке равно $d_r < N$ и количество ненулевых элементов в каждом столбце равно $d_c < M$, то код называется регулярным, в противном случае – нерегулярным.

Графически LDPC-коды можно представить с помощью графа Таннера (рис. 1).

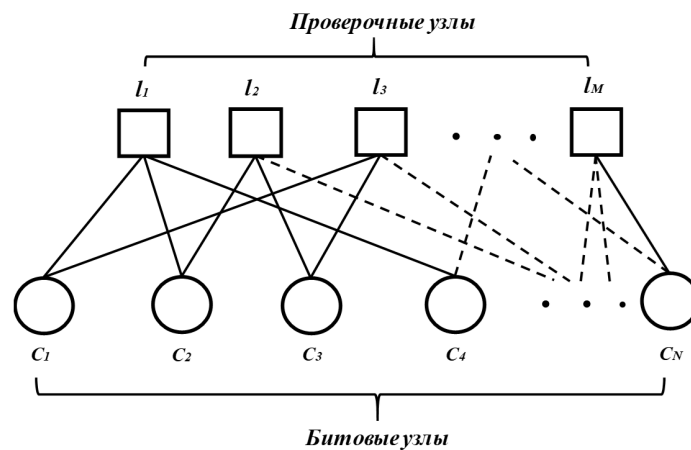


Рис. 1. Представление регулярного LDPC-кода в виде графа Таннера

На рис. 1 вершины $l_1, l_2 \dots l_M$ – проверочные узлы, а вершины $c_1, c_2 \dots c_N$ – битовые узлы. Таким образом, из графа видно, что $d_r = 3$ и $d_c = 2$.

Для декодирования LDPC-кодов используются различные алгоритмы декодирования с «жёстким» и «мягким» входом [8]. Причём «жёсткие» декодеры имеют достаточно простую конструкцию, их легко реализовать на практике, но они обладают невысокой эффективностью. А «мягкие» декодеры, наоборот, более сложные, но обладают высокой эффективностью. В данной работе рассмотрены только «мягкие» декодеры.

2.1. Алгоритм декодирования «sum-product»

Для пояснения алгоритма введём следующие обозначения:

$M_{j,i}$ – сообщение от i -го битового узла к j -му проверочному узлу; $E_{j,i}$ – сообщение от j -го проверочного узла к i -му битовому узлу; A_i – множество проверок, в которых участвует i -й символ; B_j – множество символов, которые участвуют в j -й проверке.

Алгоритм «sum-product» описан в [6, 9]. Его можно представить в виде следующих шагов:

Шаг 1. Инициализация: для каждого принятого символа c_i установить значения $M_{j,i}$:

$$M_{i,j} = L(c_i), \quad (1)$$

где $L(c_i)$ – априорная функция логарифмического отношения правдоподобия (LLR) принятого символа c_i :

$$L(c_i) = \ln \frac{p(c_i = 0)}{p(c_i = 1)}. \quad (2)$$

Шаг 2. Формирование сообщений от битовых узлов к проверочным (для первой итерации используется формула (1)):

$$M_{j,i} = L(c_i) + \sum_{j' \in A_i, j' \neq i} E_{j',i}. \quad (3)$$

Шаг 3. Формирование сообщений от проверочных узлов к битовым:

$$E_{j,i} = 2 \tanh^{-1} \left(\prod_{i' \in B_j, i' \neq i} \tanh(M_{j,i'}/2) \right). \quad (4)$$

Шаг 4. Расчёт значения общего суммарного LLR и определение новых значений битов кодового слова:

$$L(P_i) = L(c_i) + \sum_{j \in A_i} E_{j,i}. \quad (5)$$

По полученным значениям $L(P_i)$ определить новые значения битов кодового слова:

$$c_i = \begin{cases} 1, & L(P_i) \leq 0, \\ 0, & L(P_i) > 0. \end{cases} \quad (6)$$

Процесс декодирования будет остановлен тогда, когда будет найдено допустимое кодовое слово, т.е. $sH^T = 0$, или пока не будет выполнено заданное количество итераций.

2.2. Алгоритм декодирования «min-sum»

Алгоритм «min-sum» является модифицированным алгоритмом «sum-product», полученный путём упрощения вычисления сообщений от проверочных узлов к битовым. Для этого применено упрощение расчёта LLR суммы по модулю 2 множества случайных статистически независимых величин $C_1, C_2 \dots C_n$ [4]:

$$L(C_1 \oplus C_2 \dots \oplus C_n) \approx \left(\prod_{i=1}^n \text{sign}(L(C_i)) \right) \cdot \min_{i=1 \dots n} (|L(C_i)|). \quad (7)$$

Таким образом, в алгоритме «min-sum» упрощается вычисление $E_{j,i}$, считая что элемент, соответствующий наименьшему $M_{j,i'}$, является определяющим при расчёте. Поэтому $E_{j,i}$ может быть аппроксимирован следующим образом:

$$E_{j,i} \approx \left(\prod_{i' \in B_j, i' \neq i} \text{sign}(M_{j,i'}) \right) \cdot \min_{i' \in B_j, i' \neq i} (|M_{j,i'}|). \quad (8)$$

Вычисление значения $E_{j,i}$ теперь требует только простых операций поиска минимума и сложения.

3. Модифицированные алгоритмы «min-sum»

Как уже было сказано выше, из-за упрощения алгоритм «min-sum» обладает достаточно большим энергетическим проигрышем, порядка 0.2...0.5 дБ, по сравнению с «sum-product» алгоритмом. В связи с этим задача его модификации с целью повышения эффективности является актуальной. Ниже представлены модифицированные версии алгоритма «min-sum».

3.1. Алгоритм «min-sum normalized»

Отличие алгоритма «min-sum normalized» [2, 4, 6] от алгоритма «min-sum» состоит в том, что теперь к расчёту сообщений от проверочных узлов к битовым добавляется один элемент, так называемый коэффициент нормализации α для сокращения разницы в значениях, вычисляемых по формулам (4) и (8):

$$E_{j,i} = \alpha \cdot \left(\prod_{i' \in B_j, i' \neq i} \text{sign}(M_{j,i'}) \right) \cdot \min_{i' \in B_j, i' \neq i} (|M_{j,i'}|), \quad (9)$$

где значение α варьируется, как правило, в пределах (0, 1].

Для нахождения оптимального значения коэффициента α конкретного кода (в данном случае использовался регулярный LDPC-код со скоростью $R = 1/2$) было проведено исследование в среде моделирования Matlab. При этом использовалась проверочная матрица регулярного кода $H(408 \times 204)$ [10]. Вес каждого столбца: $d_c = 3$, а вес каждой строки: $d_r = 6$. Максимальное количество итераций – 5, канал связи – АБГШ, модуляция – BPSK.

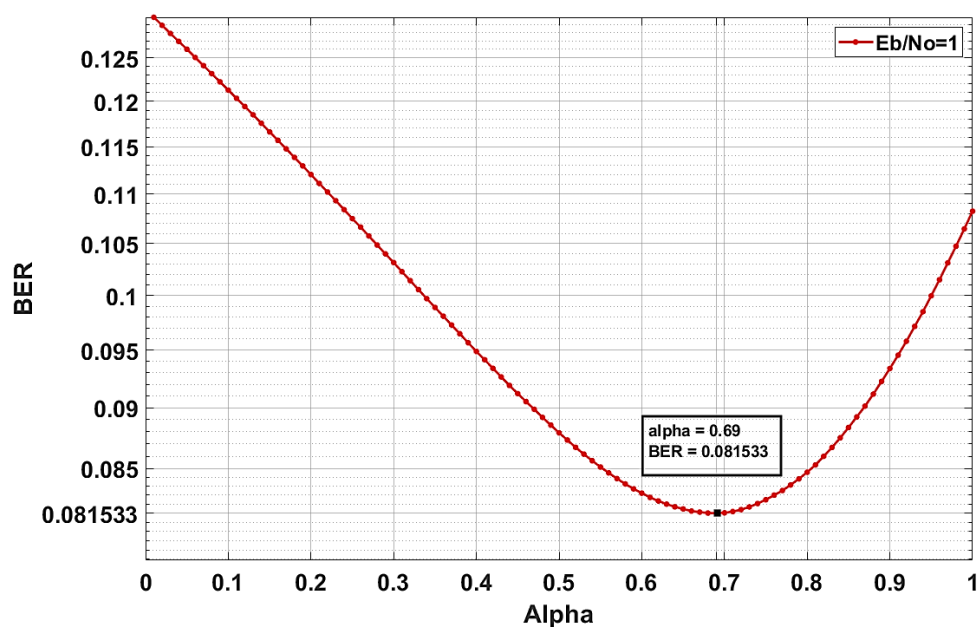


Рис. 2. Зависимость вероятности битовой ошибки (BER) от коэффициента нормализации α при $E_b/N_0 = 1$ дБ

На рис. 2 видно, что при $\alpha = 0.69$ вероятность битовой ошибки достигает минимума – 0.081533.

3.2. Алгоритм «min-sum offset»

Алгоритм «min-sum offset» [1, 2, 4] является альтернативной модификацией «min-sum» алгоритма. В отличие от алгоритма «min-sum normalized», в расчёте $E_{j,i}$ используется коэффициент сдвига β :

$$E_{j,i} = \left(\prod_{i' \in B_j, i' \neq i} \text{sign}(M_{j,i'}) \right) \cdot \max \left(\left(\min_{i' \in B_j, i' \neq i} (|M_{j,i'}|) - \beta \right), 0 \right). \quad (10)$$

Значение β также варьируется в пределах (0, 1]. При таких же условиях, как и для α , была построена зависимость для нахождения оптимального значения β . На рис. 3 показано, что значение BER достигло минимума – 0.082560 при $\beta = 0.73$.

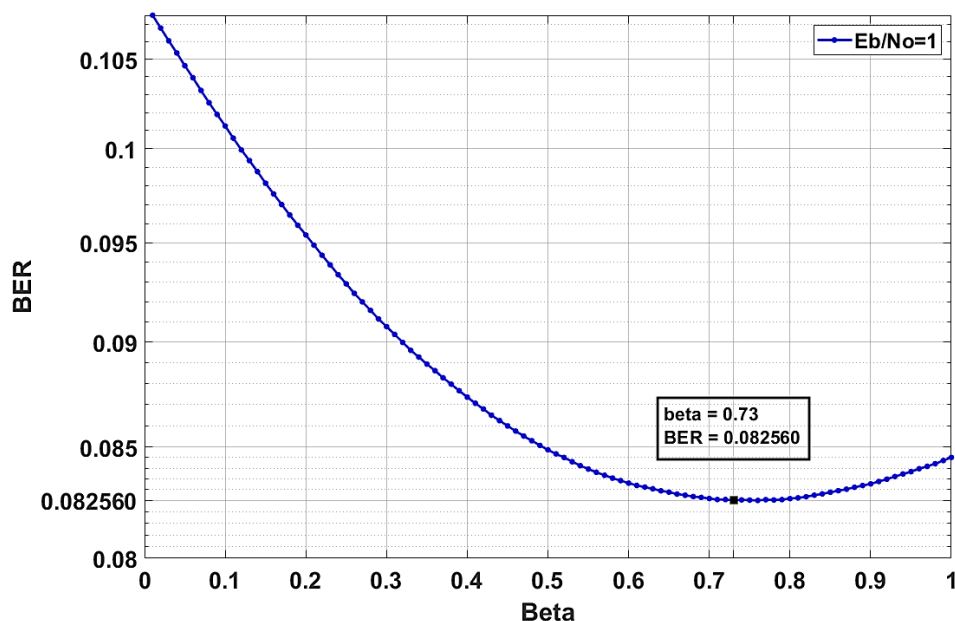


Рис. 3. Зависимость вероятности битовой ошибки (BER) от коэффициента сдвига β при $E_b/N_0 = 1$ дБ

3.3. Комбинированный алгоритм «min-sum»

Для достижения максимальной эффективности, приближенной к алгоритму «sum-product», предложен следующий вариант комбинации двух вышерассмотренных алгоритмов: «min-sum normalized» и «min-sum offset», т.е. в расчёте $E_{j,i}$ присутствуют одновременно два коэффициента α и β :

$$E_{j,i} = \alpha \cdot \left(\prod_{i' \in B_j, i' \neq i} \text{sign}(M_{j,i'}) \right) \cdot \max \left(\left(\min_{i' \in B_j, i' \neq i} (|M_{j,i'}|) - \beta \right), 0 \right). \quad (11)$$

На рис. 4 представлена поверхность, иллюстрирующая зависимость вероятности битовой ошибки от коэффициентов α и β . Можно заметить, что минимальная вероятность битовой ошибки достигается при $\alpha = 0.84, \beta = 0.44$.

В табл. 1 приведены оптимальные значения α и β для нескольких значений E_b/N_0 для различных алгоритмов декодирования.

Т а б л и ц а 1

Сравнительная характеристика помехоустойчивости различных алгоритмов декодирования

E_b/N_0	«min-sum»	«min-sum normalized»		«min-sum offset»		Комбинированный		
		α	BER	β	BER	α	β	BER
1	$0.10601 \cdot 10^{-0}$	0.69	$0.81533 \cdot 10^{-1}$	0.73	$0.82560 \cdot 10^{-1}$	0.84	0.44	$0.80552 \cdot 10^{-1}$
2	$0.41161 \cdot 10^{-1}$	0.77	$0.26561 \cdot 10^{-1}$	0.53	$0.27528 \cdot 10^{-1}$	0.81	0.22	$0.25399 \cdot 10^{-1}$
3	$0.45882 \cdot 10^{-2}$	0.81	$0.26443 \cdot 10^{-2}$	0.42	$0.27289 \cdot 10^{-2}$	0.86	0.14	$0.23823 \cdot 10^{-2}$

Из рис. 4 и табл. 1 видно, что значения вероятности ошибки на бит при комбинированном алгоритме для различных значений E_b/N_0 являются минимальными по сравнению с двумя другими модификациями «min-sum» алгоритма.

В табл. 2 представлены расчёты количества операций для выполнения одной итерации рассмотренных алгоритмов. Для конкретной проверочной матрицы $H(408 \times 204)$, имеющей вес каждого столбца: $d_c = 3$, а вес каждой строки: $d_r = 6$, получена табл. 3.

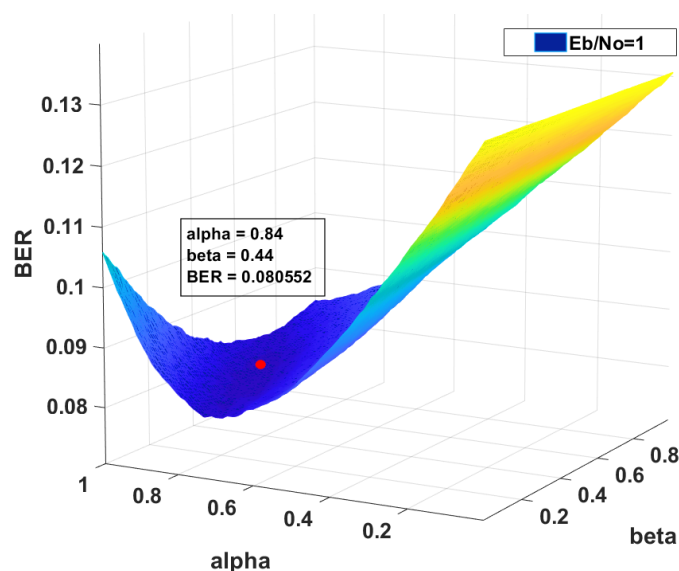


Рис. 4. Зависимость вероятности битовой ошибки (BER) от коэффициентов α и β при $E_b/N_0 = 1$ дБ

Т а б л и ц а 2

Количество вычислительных операций для выполнения одной итерации алгоритмов декодирования «min-sum», «min-sum normalized», «min-sum offset» и предложенного комбинированного алгоритма

Операция	Количество операций			
	«min-sum»	«min-sum normalized»	«min-sum offset»	«Комбинированный»
Сложение	Nd_c^2	Nd_c^2	$Nd_c^2 + Md_r$	$Nd_c^2 + Md_r$
Умножение	$Md_r(d_r - 1)$	$Md_r(d_r - 1) + Md_r$	$Md_r(d_r - 1)$	$Md_r(d_r - 1) + Md_r$
Сравнение	$Md_r(d_r - 2) + N$	$Md_r(d_r - 2) + N$	$Md_r(d_r - 1) + N$	$Md_r(d_r - 1) + N$
Взятие модуля числа	$Md_r(d_r - 1)$	$Md_r(d_r - 1)$	$Md_r(d_r - 1)$	$Md_r(d_r - 1)$
Сложение по модулю 2	Md_r	Md_r	Md_r	Md_r

Т а б л и ц а 3

Количество вычислительных операций, необходимых для выполнения одной итерации алгоритмов декодирования для LDPC-кода 408×204 [10]

Операция	Количество операций			
	«min-sum»	«min-sum normalized»	«min-sum offset»	«Комбинированный»
Сложение	3672	3672	4896	4896
Умножение	6120	7344	6120	7344
Сравнение	5304	5304	6528	6528
Взятие модуля числа	6120	6120	6120	6120
Сложение по модулю 2	1224	1224	1224	1224

Также построена гистограмма для сравнения количества операций для выполнения одной итерации различных алгоритмов декодирования (рис. 5). Из гистограммы видно,

что самым сложным является комбинированный метод, затем алгоритмы «min-sum offset», «min-sum normalized» и «min-sum».

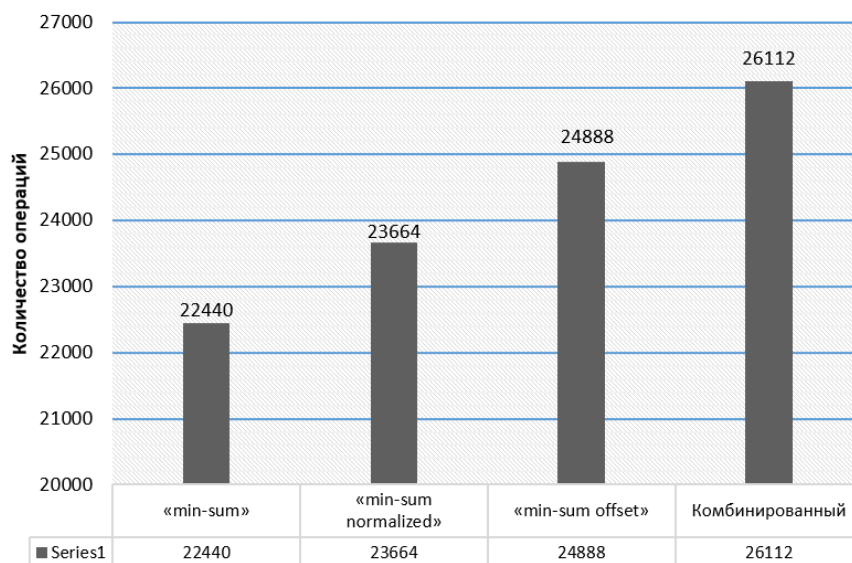


Рис. 5. Сложность различных алгоритмов декодирования

4. Заключение

В статье рассмотрены несколько алгоритмов декодирования LDPC-кодов, основанных на алгоритме «min-sum». Была исследована возможность использования двух коэффициентов (коэффициент нормализации и коэффициент сдвига) для повышения эффективности алгоритма «min-sum», получены их оптимальные значения для каждого алгоритма. А также предложен комбинированный алгоритм «min-sum», в котором использовались одновременно два коэффициента, и получены их оптимальные значения для разных значений E_b/N_0 .

Также проведено сравнение рассмотренных алгоритмов по эффективности. Следует отметить, что самым эффективным оказался предложенный комбинированный алгоритм.

Проведена оценка сложности рассмотренных алгоритмов. По общему результату исследования комбинированный алгоритм имеет лучший результат по сравнению с двумя остальными из семейства алгоритмов min-sum, но, к сожалению, его вычислительная сложность получается наибольшей.

Литература

1. Roberts M.K. Combined Normalized and Offset Min-Sum Decoding Algorithm for Irregular LDPC Codes // National Conference on Networking, Embedded and Wireless Systems (NEWS-2016), BMS College of Engineering, Bangalore, INDIA. 2016.
2. Gunnam K., Choi G. A low power architecture for min-sum decoding of LDPC codes // TAMU, ECE Technical Report, May. 2006.
3. Gallager R.G. Low-Density Parity-Check Codes // Monograph, M.I.T. Press. 1963.
4. Кирьянов И.А. Декодирование кодов с малой плотностью проверок на чётность. Диссертация на соискание ученой степени кандидата технических наук. 2015.
5. Коротков Л.Н., Башкиров А.В., Свиридова И.В. Использование LDPC-кодов // Вестник Воронежского государственного технического университета. 2013. Т. 9, № 6–3. С. 41–44.

6. *Islam M.R., Shafiullah D.S., Faisal M.M.A., Rahman I.* Optimized min-sum decoding algorithm for low density parity check codes // International Journal of Advanced Computer Science and Applications. 2011. V. 2, N 12. P. 168–174.
7. *Башкиров А.В., Хорошайлова М.В., Борисов В.И.* Реализации LDPC-декодера низкой сложности с использованием алгоритма Min-sum // Вестник Воронежского государственного технического университета. 2016. № 5. С. 82–86.
8. *Хлынов А.А.* Исследование итеративных алгоритмов декодирования кодов с низкой плотностью проверок на четность // Труды МФТИ. 2016. Т. 8, № 4. С. 13–17.
9. *Johnson S.J.* Introducing low-density parity-check codes // University of Newcastle, Australia. 2006. V. 1.
10. *MacKay D.J.C.* Encyclopedia of Sparse Graph Codes. 2014.
<http://www.inference.org.uk/mackay/codes/data.html>

References

1. *Roberts M.K.* Combined Normalized and Offset Min-Sum Decoding Algorithm for Irregular LDPC Codes. National Conference on Networking, Embedded and Wireless Systems (NEWS-2016), BMS College of Engineering, Bangalore, INDIA. 2016.
2. *Gunnam K., Choi G.* A low power architecture for min-sum decoding of LDPC codes. TAMU, ECE Technical Report, May. 2006.
3. *Gallager R.G.* Low-Density Party-Check Codes. Monograph, M.I.T. Press. 1963.
4. *Kiryanov I.A.* Decoding of codes with a low density parity checks. dissertation for the degree of candidate of technical sciences. 2015. (in Russian).
5. *Korotkov L.N., Bashkirov A.V., Sviridova I.V.* Using LDPC Codes. Vestnik of Voronezh State Technical University. 2013. V. 9, N 6–3. P. 41–44. (in Russian).
6. *Islam M.R., Shafiullah D.S., Faisal M.M.A., Rahman I.* Optimized min-sum decoding algorithm for low density parity check codes. International Journal of Advanced Computer Science and Applications. 2011. V. 2. N 12. P. 168–174.
7. *Bashkirov A.V., Khoroshaylova M.V., Borisov V.I.* Implementations of LDPC decoder of low complexity using Min-sum algorithm. Vestnik of Voronezh state technical University. 2016. N 5. P. 82–86. (in Russian).
8. *Khlynov A.A.* Investigation of iterative algorithms for decoding codes with a low density parity checks. Proceedings of MIPT. 2016. V. 8, N 4. P. 13–17. (in Russian).
9. *Johnson S.J.* Introducing low-density parity-check codes. University of Newcastle, Australia. 2006. V. 1.
10. *MacKay D.J.C.* Encyclopedia of Sparse Graph Codes. 2014.
<http://www.inference.org.uk/mackay/codes/data.html>

Поступила в редакцию 05.10.2020