

# Использование Discrete Density Evolution при конструировании матриц сверточных кодов\*

Угловский Артём<sup>1,2</sup>, Крещук Алексей<sup>1,3</sup>

<sup>1</sup> Институт проблем передачи информации РАН

<sup>2</sup> Московский физико-технический институт (НИУ)

<sup>3</sup> Высшая школа экономики (НИУ)

Email: [uglovskii@wireless.iitp.ru](mailto:uglovskii@wireless.iitp.ru)

**Аннотация** — В данной статье рассматривается применение разработанного метода конструирования матриц сверточных LDPC кодов с помощью Discrete Density Evolution (далее DDE). До сих пор конструирование сверточных протоматриц выглядело следующим образом: сначала применяем DDE к блочной протоматрице, тем самым получая лучшую, а затем из нее случайным образом конструируем сверточную протоматрицу. Возникло предположение, что эта случайность не дает хорошего результата, которого хотелось бы достичь. И именно по этой причине появилась необходимость в усовершенствовании метода конструирования сверточных протоматриц. Новый способ должен позволить выбрать лучшую сверточную протоматрицу из набора случайно сгенерированных, т.е. матрицу, имеющую наименьший порог (threshold) на графике зависимости bit error rate (далее BER) от signal to noise ratio (далее SNR). Таким образом, удастся подобрать более подходящую под условия канала матрицу, и, тем самым, улучшить эффективность декодера.

**Ключевые слова** — Сверточные Low-density parity-check (LDPC) коды, Discrete Density Evolution (DDE), протоматрица

---

\* Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ

## 1 Введение

LDPC коды [3] — это линейные коды с сильно разреженной матрицей проверки на четность (Low Density Parity Check). В настоящий момент LDPC коды пользуются большой популярностью в различных сферах кодирования, в том числе и в 5G [7], из-за лучшего по сравнению с другими кодами показателя BER, благодаря чему удается достичь более совершенных характеристик декодирования. Выделяют блочные и сверточные LDPC коды. Последние демонстрируют более высокую производительность [4], и в данной статье речь пойдет именно о сверточных LDPC кодах.

Выбор удачной протоматрицы для сверточных LDPC кодов — важная задача, так как хорошая протоматрица улучшает производительность и эффективность декодера, которые напрямую связаны с параметрами канала.

Для оценки матрицы используют несколько параметров: порог и полку (error floor). Порог определяет значение SNR, при которой декодер начинает «работать», то есть достигается допустимое значение BER. Чем меньше порог, тем более устойчив декодер, и небольшие изменения SNR канала не приведут к сильному изменению BER. Полка показывает вероятность ошибочного декодирования, когда порог уже пройден.

Дальнейшее изложение организовано следующим образом: в секции 2 представлено описание алгоритмов, используемых в работе, секция 3 приводит результаты проделанных экспериментов, в секции 4 содержатся выводы о проделанной работе.

Таблица 1. Список сокращений

LDPC	Low-Density Parity-Check
BER	Bit Error Rate
SNR	Signal to Noise Ratio
DE	Density Evolution
DDE	Discrete Density Evolution
LLR	Log-Likelihood Ratio
$R_{cv}$	сообщение от узла проверки к узлу переменной
$Q_{vc}$	сообщение от узла переменной к узлу проверки
BP	Belief Propagation

## 2 Описание модели

### 2.1 Конструирование сверточной протоматрицы

Для преобразования блочной матрицы в сверточную будем пользоваться пространственным связыванием (Spatial Coupling [5]). Суть данного метода состоит в представлении блочной протоматрицы  $B$  суммой  $B = \sum_{i=0}^{\infty} B_i$ , в которой  $B_i$  не имеют единиц на одинаковых позициях. Далее из этих матриц составляется сверточная протоматрица  $H$  в виде лесенки (Spatially Coupled LDPC), как на рис.1.

$$\mathbf{H} = \begin{bmatrix} \mathbf{B}_0 & & & & & & \\ & \mathbf{B}_1 & & & & & \\ & \vdots & & & & & \\ & & \mathbf{B}_1 & & \ddots & & \\ \mathbf{B}_{w_0} & & \vdots & & \ddots & & \mathbf{B}_0 \\ & & & \mathbf{B}_{w_0} & & & \mathbf{B}_1 \\ & & & & & \ddots & \vdots \\ & & & & & & \mathbf{B}_{w_0} \end{bmatrix}$$

Рис. 1. сверточная протоматрица

## 2.2 Декодирование сверточных кодов

С алгоритмом декодирования и принципом обмена сообщениями в двудольном графе Таннера можно ознакомиться, прочитав соответствующие материалы [2]. Мы же полагаем, что читатель знаком с основами декодирования.

Сверточный код, в отличие от блочного, обладает памятью. Из-за его полубесконечности не получится сначала обновить все R-сообщения, а потом все Q-сообщения (все сообщения представляют собой LLR). Вместо этого для каждого проверочного узла мы полностью обновляем ребра, входящие в него, то есть и R- и Q-сообщения (layered schedule [8]). При этом, очевидно, не все сообщения будут от одной итерации, некоторые из них будут от предыдущей. Если внимательно посмотреть на рис.2, то можно заметить, что блоки переменных, находящиеся левее первой синей черты, не участвуют в расчете блока проверочных узлов, находящихся между красными линиями. Это значит, что для проверочных узлов выше красного блока можно начинать следующую итерацию. На этом же рисунке наглядно показано, что часть сообщений (которые под красным блоком) от переменных берутся с предыдущей итерации. Условие остановки задается количеством итераций, так как синдром для сверточного кода проверить нельзя.

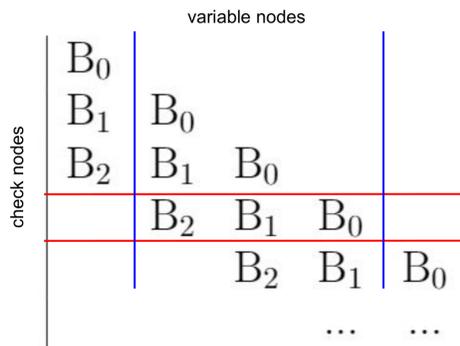


Рис. 2. Построчное декодирование

### 2.3 MinSum

Сам алгоритм декодирования построен на обмене сообщениями между проверочными узлами и узлами переменных. На графе Таннера эти сообщения можно поместить на ребра и обозначить как  $R_{cv}$  и  $Q_{vc}$  (на каждом ребре оба сообщения). За одну итерацию мы обновляем  $R_{cv}$ -сообщения, зная  $Q_{vc}$ , а затем пересчитываем  $Q_{vc}$ , имея обновленные  $R_{cv}$ .

Для обновления сообщений мы используем метод *MinSum* [1], [6], который представлен формулами (1)-(2), где  $L_c(v_i)$  — это LLR, полученный из канала.

$$R_{cv} = \prod_{v' \neq v} \text{sign}(Q_{v'c}) \min_{v' \neq v} |Q_{v'c}|. \quad (1)$$

$$Q_{vc} = L_c(v_i) + \sum_{c' \neq c} R_{c'v} \quad (2)$$

При реализации декодера правую часть (1) часто домножают на коэффициент  $\beta = 0,75$  для большей точности.

### 2.4 DDE для сверточных кодов

При использовании DDE значения LLR-сообщений являются дискретными и ограниченными. Максимальное и минимальное значения, которые представляют собой  $+\infty$  и  $-\infty$  соответственно, вычисляются исходя из количества бит, отведенных под одно сообщение. Так, при 8 битах максимальным будет LLR, равный 127, а минимальным — -128. В нашей реализации значения являются целыми числами.

В отличие от обычного процесса декодирования, где на вход подаются сами LLR, для DDE будут поступать распределения этих LLR из АБГШ (аддитивный белый гауссовский шум) канала. В связи с этим все операции с числами преобразуются в операции с распределениями (3), (4).

$$p(x = \alpha + \beta) = \sum_{\alpha_i + \beta_j = x} p(\alpha_i) * p(\beta_j) \quad (3)$$

$$p(x = \min(\alpha, \beta)) = p(\alpha \geq x) * p(\beta \geq x) - p(\alpha > x) * p(\beta > x). \quad (4)$$

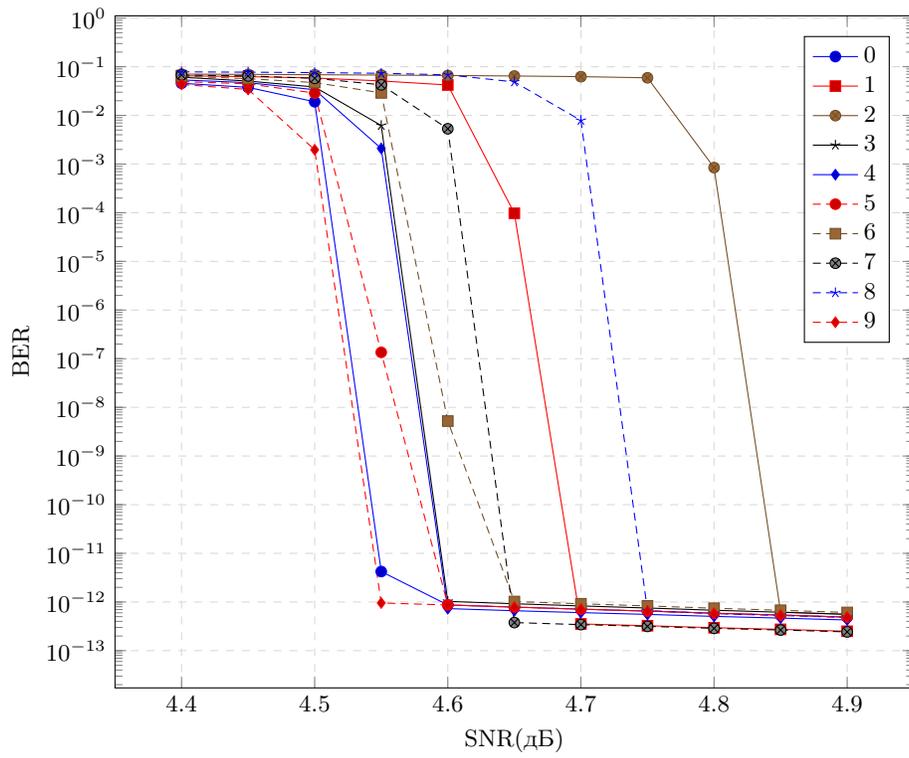
После выполнения установленного количества итераций для каждой переменной считается BER по формуле (5) и строится график зависимости BER от SNR.

$$\text{ber} = \sum_{\alpha_i < 0} p(\alpha_i). \quad (5)$$

## 3 Результаты

Генерируя случайным образом сверточные протоматрицы из заранее подобранной блочной матрицы и применяя к ним DDE на 32 итерации с 9-битным декодером, получаем график зависимости BER от SNR, изображенный на рис.3.

Рис.3 наглядно показывает разницу между случайно сгенерированными сверточными протоматрицами. Различия порогов достаточно большие (разность в 0.05 дБ является существенной для декодера). Значения полог почти одинаковы



**Рис. 3.** Сравнение сверточных протограмм с помощью DDE

и находятся в пределах нормы. Это указывает на хорошо выбранную блочную протоматрицу.

Однако представленное на рис.4 полное моделирование для матриц под номерами 9 и 2 на рис.3 (с наименьшим и наибольшим порогами соответственно) показывает, что графики зависимости BER от SNR для них практически не отличаются. Значит, данный алгоритм не работает и требует дальнейшего усовершенствования.

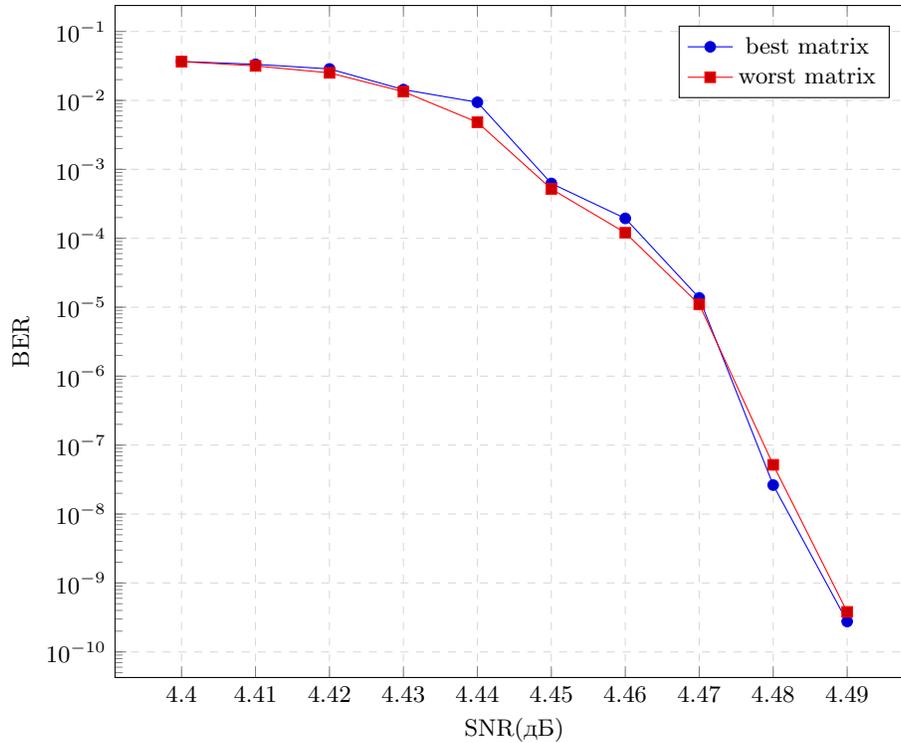


Рис. 4. Полное моделирование

## 4 Выводы

В данной статье мы рассмотрели новый подход к конструированию сверточных протоматриц с помощью DDE. Путем обмена и обновления сообщений между проверочными узлами и узлами переменных мы старались предсказать качество соответствующей протоматрицы. Здесь в роли сообщений выступают распределения LLR, а не сами LLR, как при обычном декодировании. Тем не менее, полное моделирование показало, что данный алгоритм не дает существенных результатов в выборе подходящей матрицы. Причиной этого, возможно, являются неверно выбранные параметры декодера и сообщений, а также неподходящий порядок

обновления этих сообщений. Одними из предполагаемых способов улучшения алгоритма являются уменьшение сообщения (scaling) при подсчете *MinSum*, что может увеличить точность, и переход к другому типу обновления, например, к потоковому (flooding schedule) с адаптацией его к сверточному коду. Дальнейшее исследование будет продолжаться в надежде найти способ, позволяющий решить проблему выбора удачной протоматрицы без полного моделирования.

## Список литературы

- [1] J. Chen и M.P.C. Fossorier. “Density evolution for two improved BP-Based decoding algorithms of LDPC codes”. в: *IEEE Communications Letters* 6.5 (2002), с. 208–210. DOI: 10.1109/4234.1001666.
- [2] M.P.C. Fossorier, M. Mihaljevic и H. Imai. “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation”. в: *IEEE Transactions on Communications* 47.5 (1999), с. 673–680. DOI: 10.1109/26.768759.
- [3] R. Gallager. “Low-density parity-check codes”. в: *IRE Transactions on Information Theory* 8.1 (1962), с. 21–28. DOI: 10.1109/TIT.1962.1057683.
- [4] Najeeb ul Hassan, Michael Lentmaier и Gerhard P. Fettweis. “Comparison of LDPC block and LDPC convolutional codes based on their decoding latency”. в: *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. 2012, с. 225–229. DOI: 10.1109/ISTC.2012.6325232.
- [5] David G. M. Mitchell, Michael Lentmaier и Daniel J. Costello. “Spatially Coupled LDPC Codes Constructed From Protographs”. в: *IEEE Transactions on Information Theory* 61.9 (2015), с. 4866–4889. DOI: 10.1109/TIT.2015.2453267.
- [6] Seho Myung и др. “Offset and Normalized Min-Sum Algorithms for ATSC 3.0 LDPC Decoder”. в: *IEEE Transactions on Broadcasting* 63.4 (2017), с. 734–739. DOI: 10.1109/TBC.2017.2686011.
- [7] P. Vigneswari и S. Sivakumari. “Performance Analysis of Iterative minsum Message Passing Decoding Algorithm for 5G NR LDPC codes”. в: *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. 2021, с. 142–146. DOI: 10.1109/ICICV50876.2021.9388463.
- [8] Juntan Zhang и M. Fossorier. “Shuffled belief propagation decoding”. в: *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002*. т. 1. 2002, 8–15 vol.1. DOI: 10.1109/ACSSC.2002.1197141.