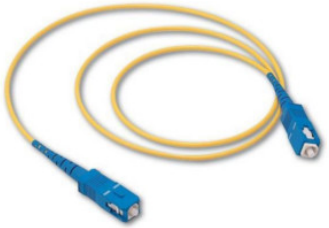
Коды на графах

К. Струминский[1]

[1] Факультет компьютерных наук
ВШЭ

Декабрь, 2015

При передаче данных возникают ошибки



Первые шаги к постановке задачи

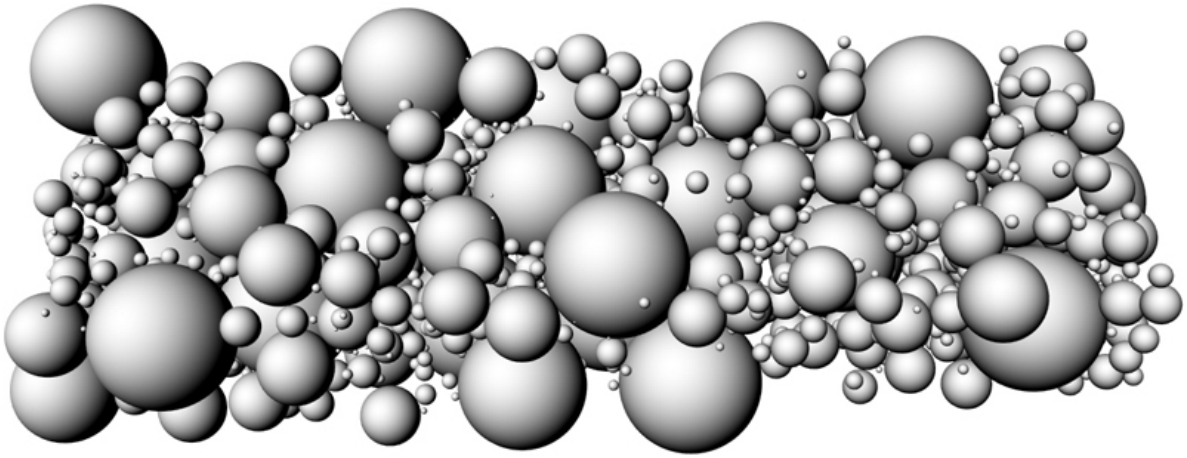
1. Сообщения из символов конечного алфавита (далее ограничимся $\mathbb{Z}_2$)
2. Для декодирования необходима избыточная информация
3. Фиксирована длина сообщений n
4. Фиксирована длина кодовых сообщений k
5. Что дальше?

Как строить коды? Отвечает Хэмминг

Буквальное понимание проблемы: код должен исправлять как можно больше ошибок.

Расстояние Хэмминга: $d(x, y) = \#\{i : x_i \neq y_i\}$

Хороший код максимизирует минимальное расстояние между словами, задача об упаковке шаров.



Как строить коды? Отвечает Шеннон

1. Ошибки вносятся каналом связи случайно и независимо
2. Для описания канала достаточно задать условное распределение $p(y|x)$
3. Для каждого канала определяется пропускная способность C
4. *теорема Шеннона* $\rightarrow$: *Коды с* $\frac{n}{k} < C$ *могут исправлять почти все ошибки*
5. *теорема Шеннона* $\leftarrow$: *А коды с* $\frac{n}{k} > C$ *не могут*

Теорема Шеннона в более строгой форме

Назовем кодирование n, k-надежным, если вероятность правильного декодирования зашумленного сообщения превышает $(1 - \varepsilon)$.

Теорема

1. *Для любых* $0 < \varepsilon < 1$ *и* $\delta > 0$ *при всех достаточно больших* k, *для любого* $(1 - \varepsilon)$ *надежного* n, k*-кодирования выполнено* $\frac{n}{k} < C + \delta$.
2. *Для любого* $\varepsilon > 0$ *существует такое* c, *что для всех* $k > 0$ *существует* $(1 - \varepsilon)$*-надежное* n, k*-кодирование с* $n > Ck - c\sqrt{k}$.

Код Хэмминга

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Код Хэмминга

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

1. Код может определяться линейным отображением сообщений G
2. В этом случае код - линейное пространство
3. Линейное пространство можно задать как ядро оператора H, матрица оператора определяется проверки на четность

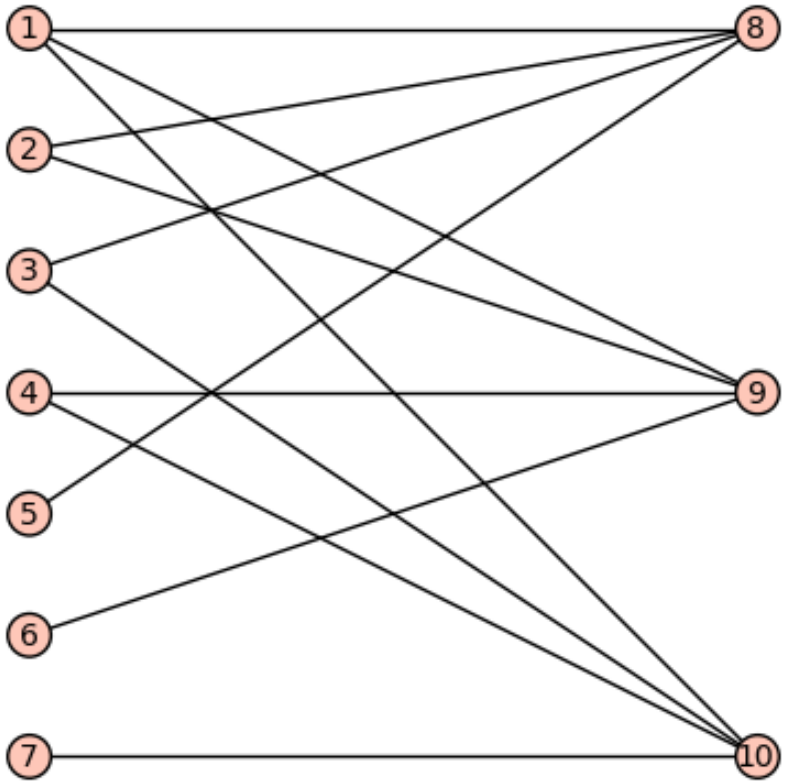
От проверок на четность к графам

$$G := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Проверки на четность задают локальные ограничения на кодовые сообщения. Их можно представить с помощью фактор-графа, представления не единственны.

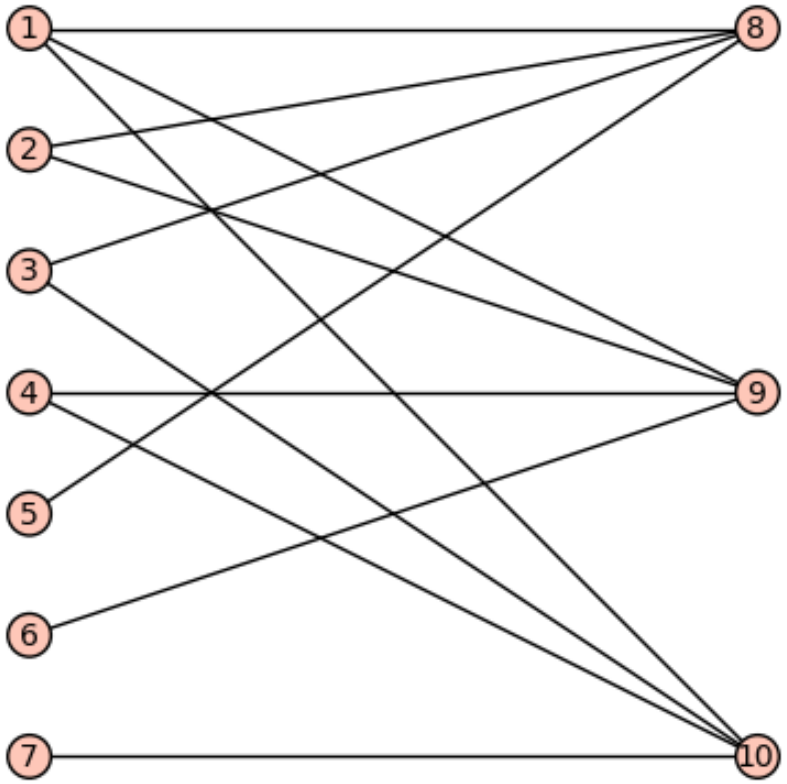
Фактор-графы

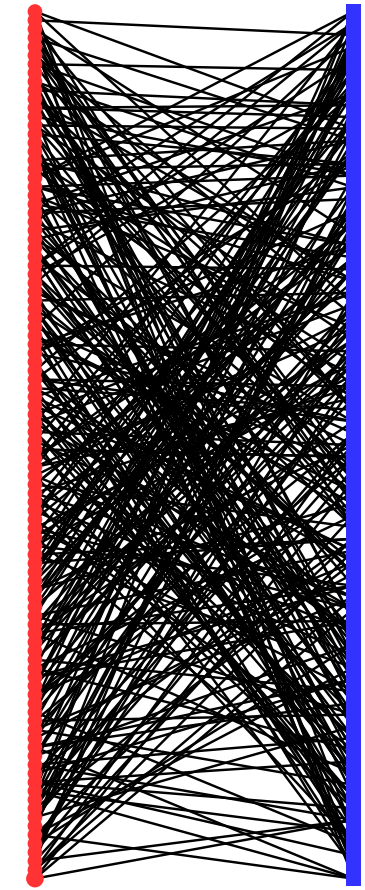
Фактор граф для кода Хэмминга и LDPC-кода.



Фактор-графы

Фактор граф для кода Хэмминга и LDPC-кода.





Что нам дают графы?

Строя код, мы получаем быстрый алгоритм декодирования.

Что нам дают графы?

Строя код, мы получаем быстрый алгоритм декодирования.

Что нам дают графы?

Строя код, мы получаем быстрый алгоритм декодирования.

1. Sum-product алгоритм (он же Belief propagation): нам заданы веса кодовых слов $w(x)$. В каждой вершине алгоритм вычисляет вес переменной: $w(\hat{x}_i) = \sum_{x \in C, x_i = \hat{x}_i} w(x)$

Что нам дают графы?

Строя код, мы получаем быстрый алгоритм декодирования.

1. Sum-product алгоритм (он же Belief propagation): нам заданы веса кодовых слов $w(x)$. В каждой вершине алгоритм вычисляет вес переменной: $w(\hat{x}_i) = \sum_{x \in C, x_i = \hat{x}_i} w(x)$
2. Если взять $w(x) = \prod p(y_i|x_i)$, получим апостериорное распределение на биты кодового сообщения при известном зашумленном $p(x_i|y)$.

Что нам дают графы?

Строя код, мы получаем быстрый алгоритм декодирования.

1. Sum-product алгоритм (он же Belief propagation): нам заданы веса кодовых слов $w(x)$. В каждой вершине алгоритм вычисляет вес переменной: $w(\hat{x}_i) = \sum_{x \in C, x_i = \hat{x}_i} w(x)$
2. Если взять $w(x) = \prod p(y_i|x_i)$, получим апостериорное распределение на биты кодового сообщения при известном зашумленном $p(x_i|y)$.
3. Можно считать в $(min, +)$ кольце с $w(x) = -\sum \log p(y_i|x_i)$, получим минимум функции правдоподобия $min_{x \in X, x_i = \hat{x}_i} p(y|x)$

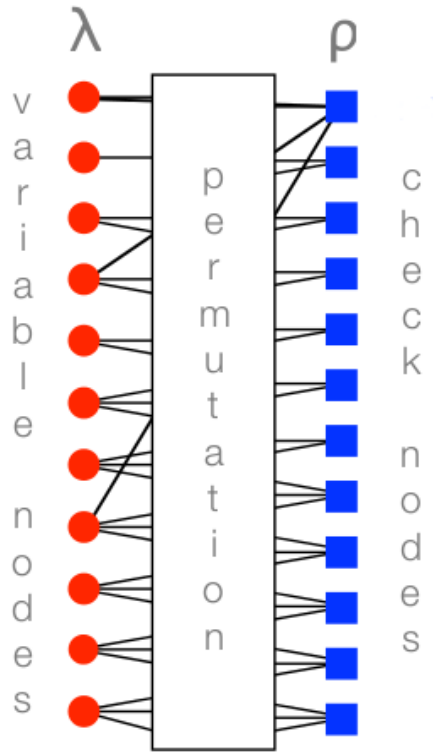
Но алгоритм работает корректно только на графах без циклов. Для графов с циклами он дает хорошее приближение искомых вероятностей.

[1]Обзор темы и хорошее изложение основ можно найти в статье Forney, G.D., Jr., "Codes on graphs: normal realizations,"in Information Theory, IEEE Transactions on , vol.47, no.2, pp.520-548, Feb 2001.

LDPC код

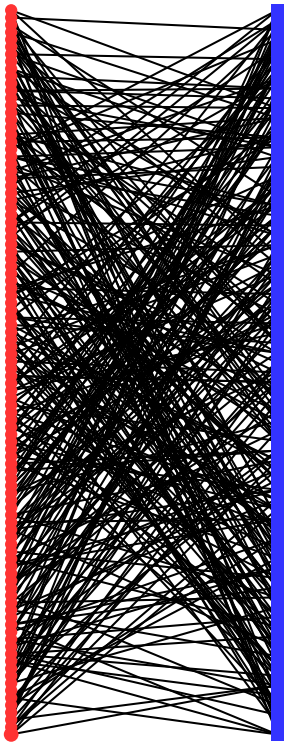
Реже граф - меньше циклов - лучше работа sum-product алгоритма.

LDPC (Low density parity check) код использует случайный двудольный граф с малыми степенями у вершин для определения кода. Для декодирования используется sum-product алгоритм.

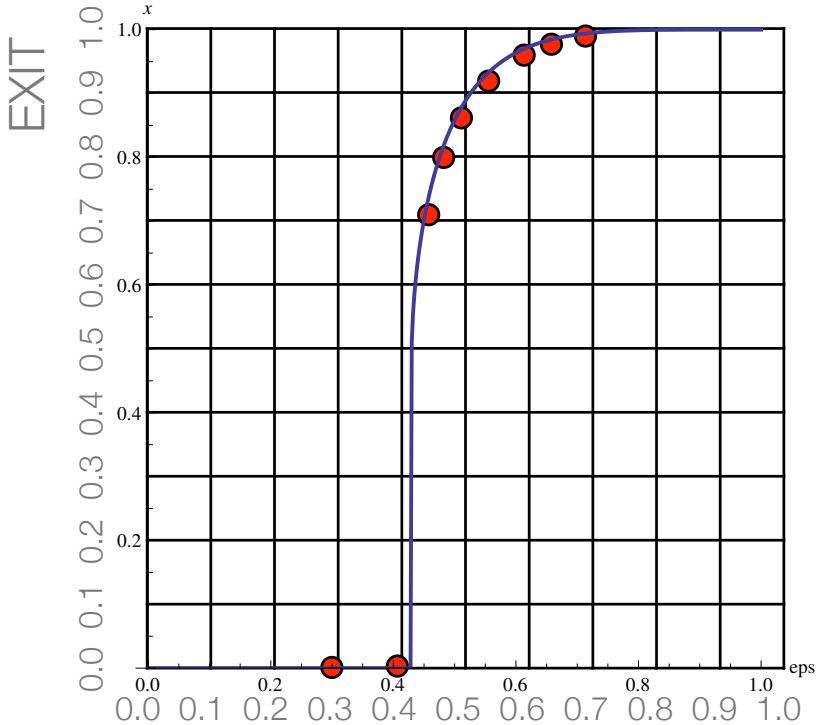


Декодирование: экспериментальные точки

Кривая для стирающего канала (Binary erasure channel, BEC), вероятность ошибки при декодировании в зависимости от вероятности стирания ϵ



(3, 6) ensemble



Кривая по экспериментальным точкам

Точка выхода кривой из нуля - *порог*, характеристика алгоритма декодирования и кода.

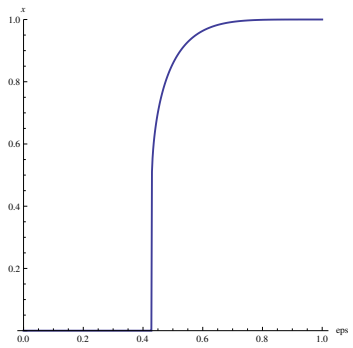
Forward Fixed points of DE

$$x^{(\ =0)} = \epsilon$$

$$y^{(\ )} = 1 - (1 - x^{(\ -1)})^{d_r - 1}$$

$$x^{(\ )} = \epsilon (y^{(\ )})^{d_l - 1}$$

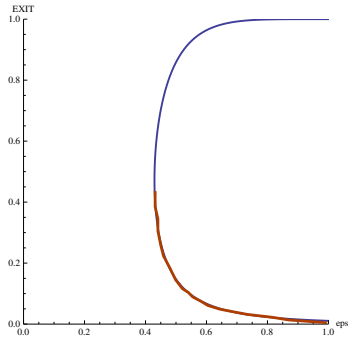
$$x = \epsilon (1 - (1 - x))^{d_r - 1})^{d_l - 1}$$



All Fixed points of DE

$$x = \epsilon (1 - (1 - x))^{d_r - 1})^{d_l - 1}$$

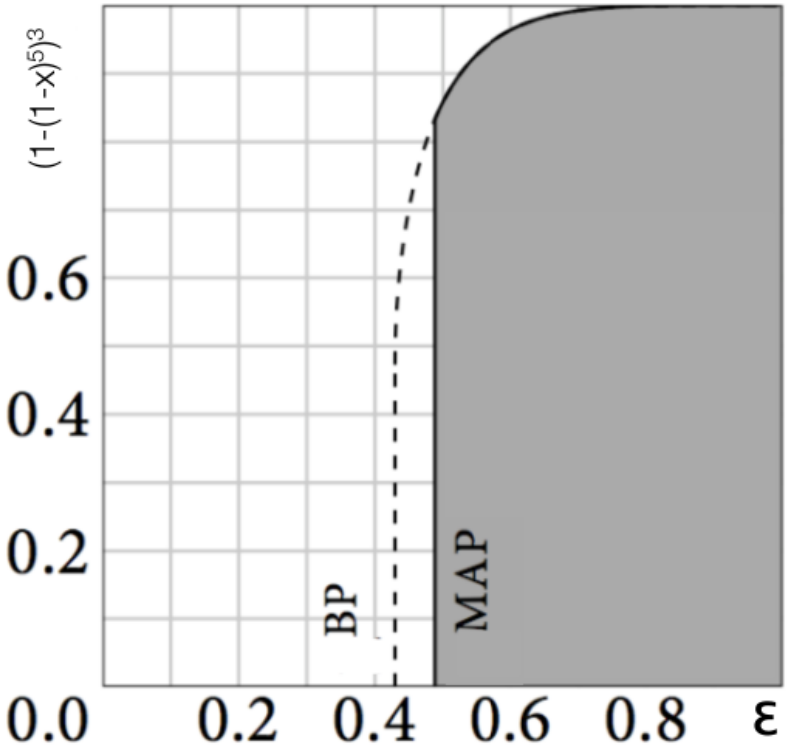
$$\epsilon = \frac{x}{(1 - (1 - x))^{d_r - 1})^{d_l - 1}}$$



[2] Вывод см. T. Richardson, R. Urbanke, "Modern Coding Theory Cambridge University Press, 2008

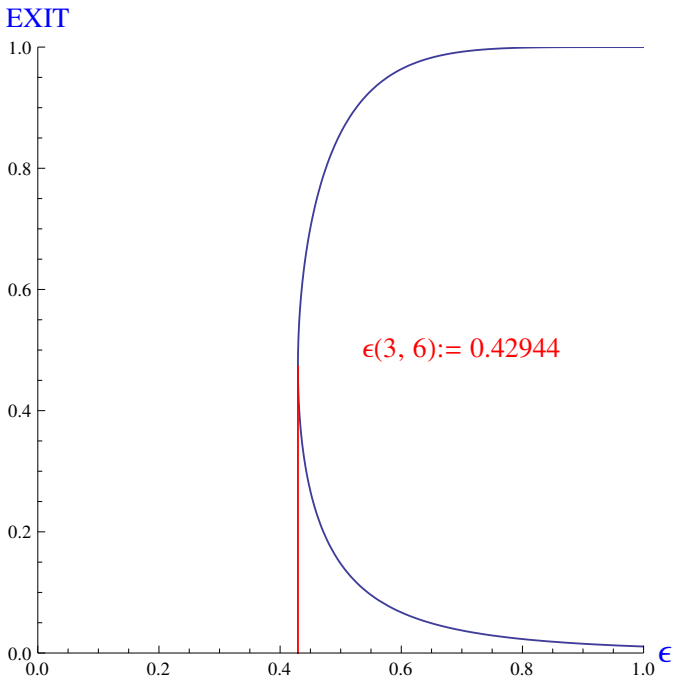
BP не MAP

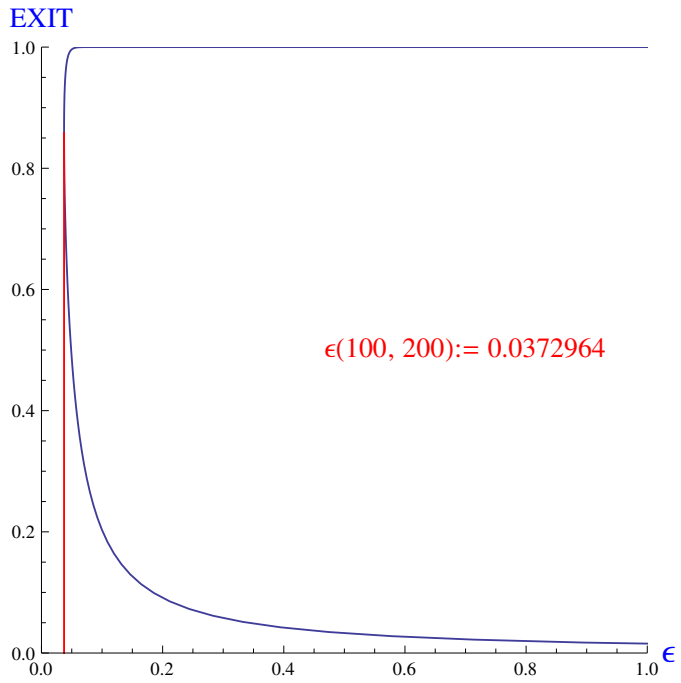
Аналитически можно вывести вероятность ошибки декодирования при вычислении истинного апостериорного распределения, в некоторых регионах она ниже ошибки при BP-декодировании.



Куда дальше?

Следующий шаг - повышение степеней вершин. Он должен привести к увеличению кодового расстояния LDPC-кодов, к повышению их качества. Но наивный подход не работает.

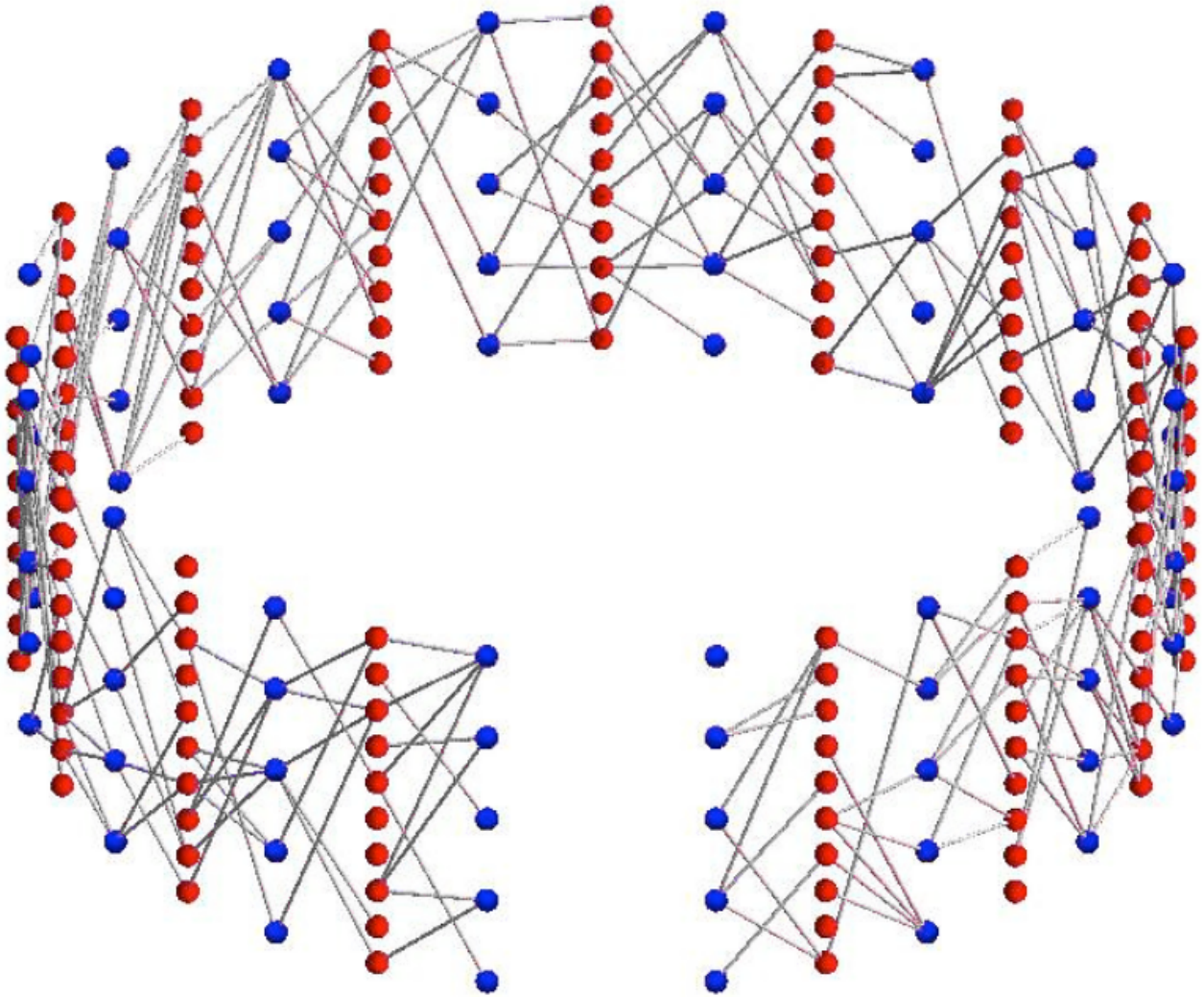




3

Картинки из доклада "Spatial Coupling – What is it, why does it work, and what are the open challenges? прочитанного R. Urbanke на EASIT 2014.

Spatially coupled LDPC



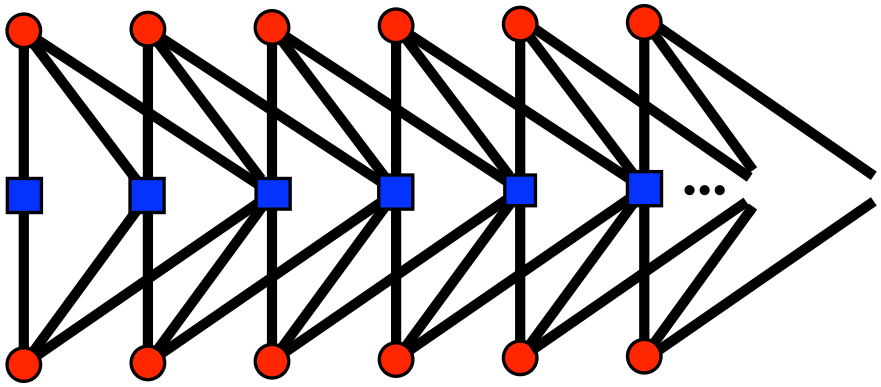
Spatially coupled LDPC

1. Граф SC-LDPC - последовательно соединенные LDPC-графы
2. Соединения также строятся случайно
3. Декодируем каждый LDPC-код слева направо, игнорируя графы справа
4. Степень вершин повышается благодаря ребрам, которые идут в LDPC-графы справа

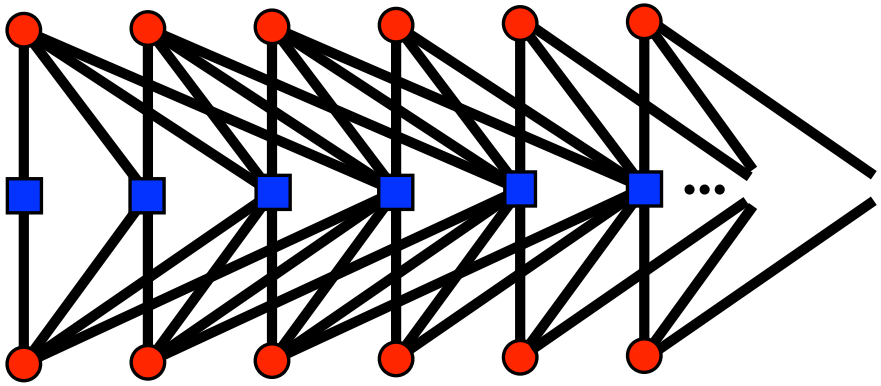
[4]Первое появление Jimenez Felstrom, A.; Zigangirov, K.S., "Time-varying periodic convolutional codes with low-density parity-check matrix,"in Information Theory, IEEE Transactions on , vol.45, no.6, pp.2181-2191, Sep 1999

Несколько ансамблей

(6, 12)-regular coupled ensemble



(8, 16)-regular coupled ensemble



Пара слов о SC-LDPC

1. Для стирающих каналов средний порог ϵ_{BP} графов "в связке"превосходит порог графов самих по себе
2. Даже приближается к порогу ϵ_{MAP}
3. Увеличение степеней вершин позволяет улучшить показатели кода