Генетический алгоритм для поиска самых лучших матриц LD(P+M)C кода

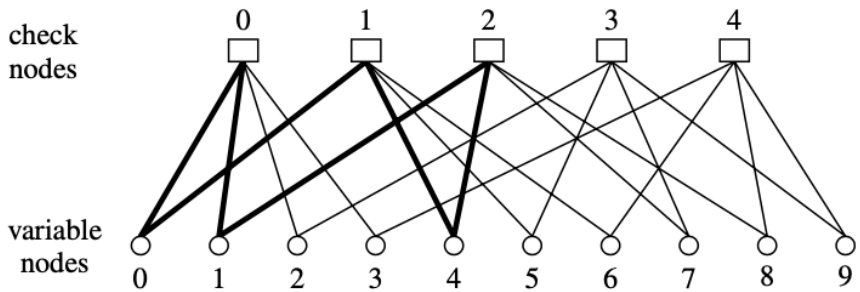
Ваня Тунёв Наташа Доброхотова-Майкова

21 августа 2020 г.

Напоминание про LDPC коды

- В проверочной матрице мало единиц
- Можно представить с помощью графа Таннера
- Есть алгоритм декодирования SUM-PRODUCT
- Сложно искать матрицы

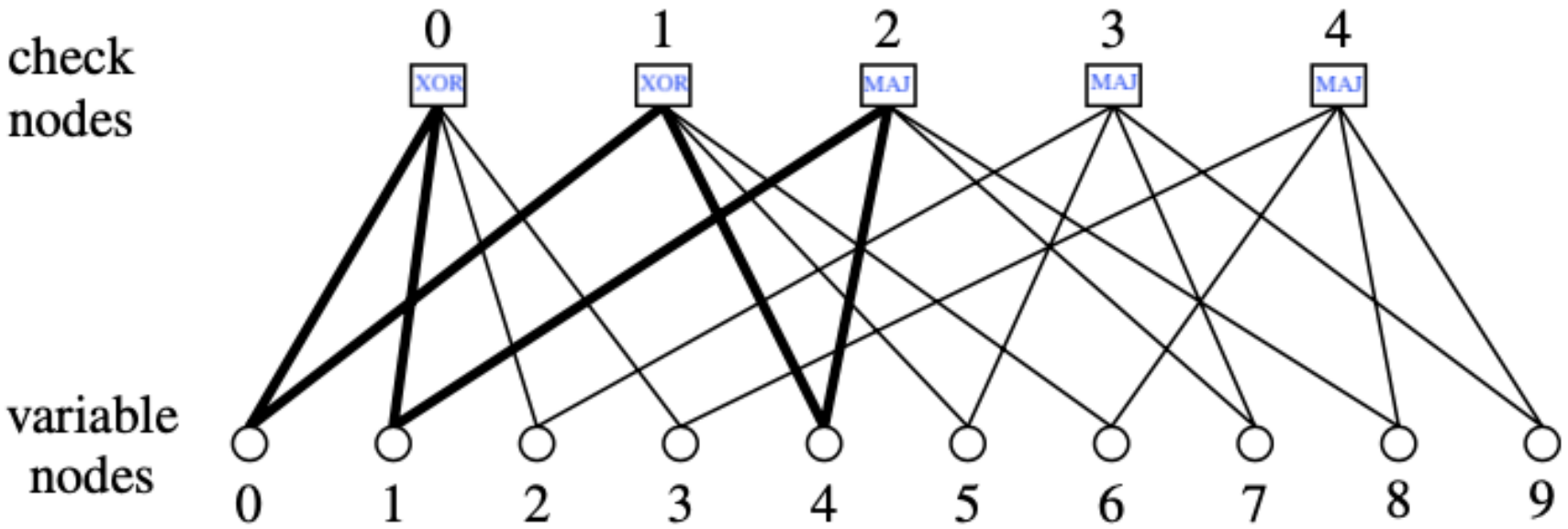
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$



Откуда появилось +M?

$$XOR(x_1, x_2, ..., x_n) = \begin{cases} 1, & \sum_{i=1}^{n} x_i \equiv 1 (mod 2) \\ 0, & \sum_{i=1}^{n} x_i \equiv 0 (mod 2) \end{cases}$$

$$MAJ(x_1, x_2, ..., x_n) = \begin{cases} 1, & \sum_{i=1}^{n} x_i \geqslant \frac{n}{2} \\ 0, & \sum_{i=1}^{n} x_i < \frac{n}{2} \end{cases}$$



Две части проекта

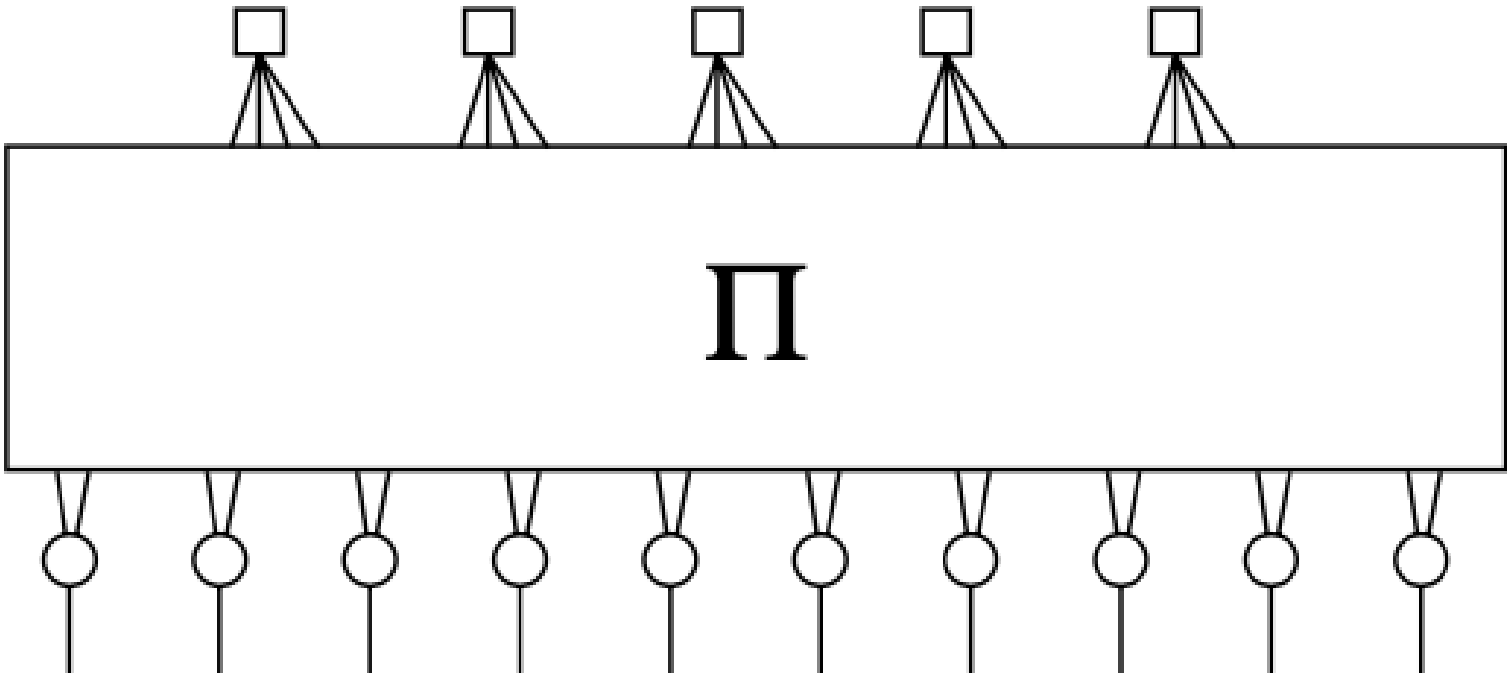
1. Добавляем нелинейности (Наташа)
2. Ищем хорошие матрицы (Ваня)

Две части проекта

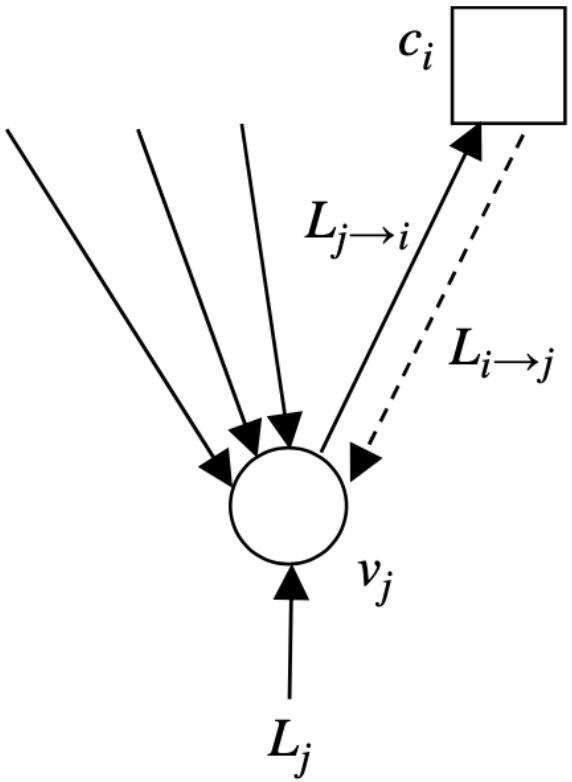
1. Добавляем нелинейности (Наташа)
2. Ищем хорошие матрицы (Ваня)

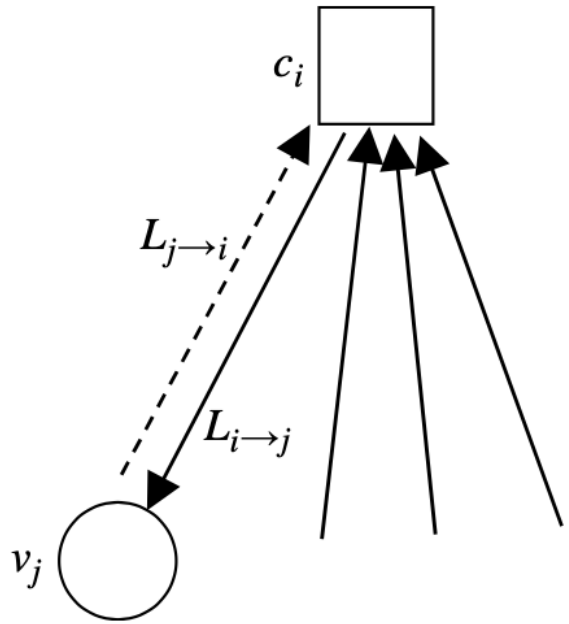
Sum Product

В основе лежит обмен сообщениями (Message Passing)
На Sum product можно смотреть как на несколько декодеров



Sum Product





Sum product с MAJ => нужен новый декодер

Пока самый простой, без всяких гиперболических тангенсов

Пока какой-то странный баг (или нет), к нему странный костыль

Две части проекта

1. Добавляем нелинейности (Наташа)

2. Ищем хорошие матрицы (Ваня)

Про генетические алгоритмы

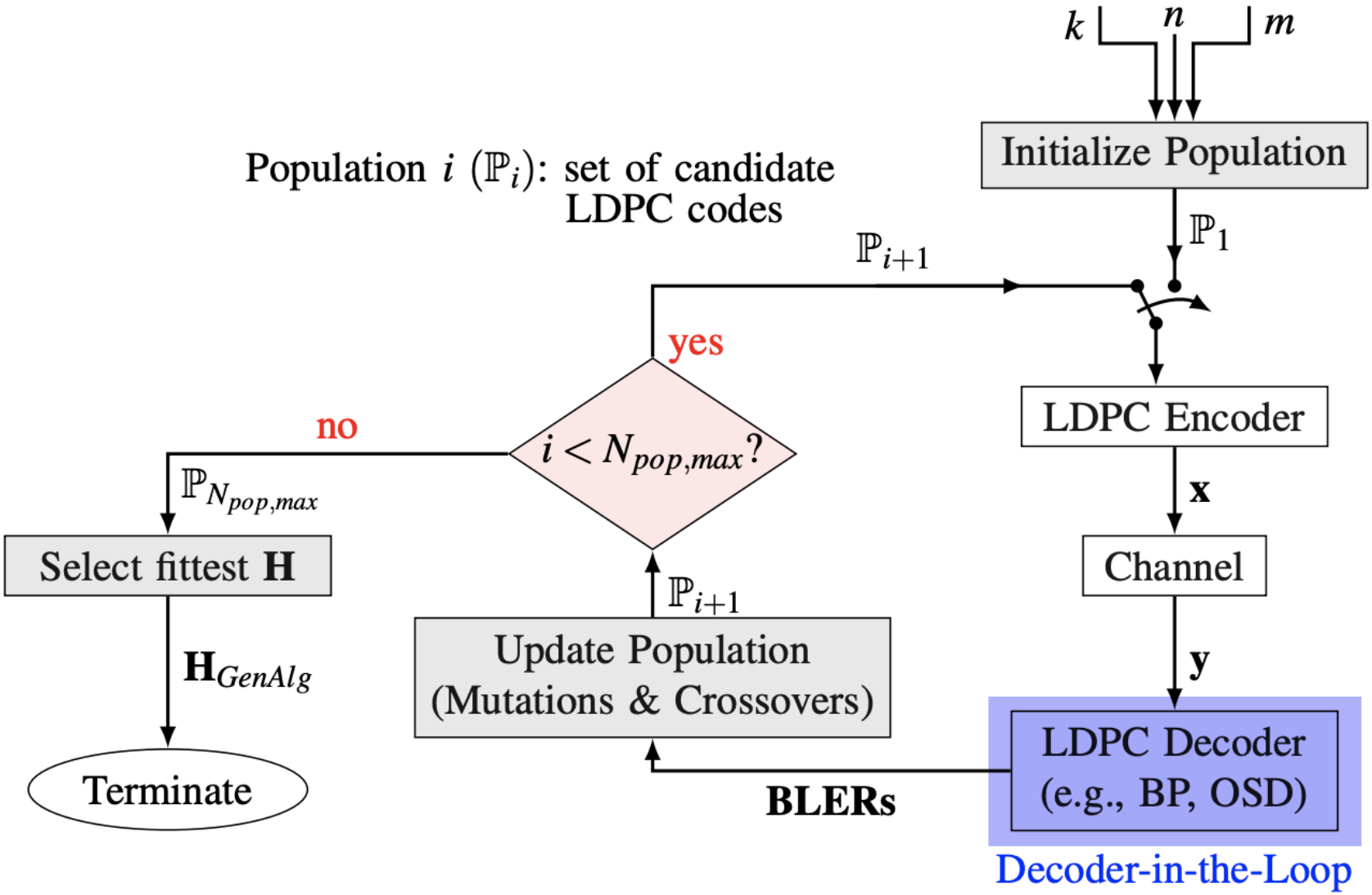
Генетические алгоритмы – алгоритмы оптимизации путём случайного подбора, комбинирования, т.е используя механизмы, похожие на эволюцию

- Создание начальной популяции
- Селекция, выбор родителей
- Мутации и скрещивания
- Анализ и оценка полученного
- Продолжать до тех пор, пока не получим желаемое

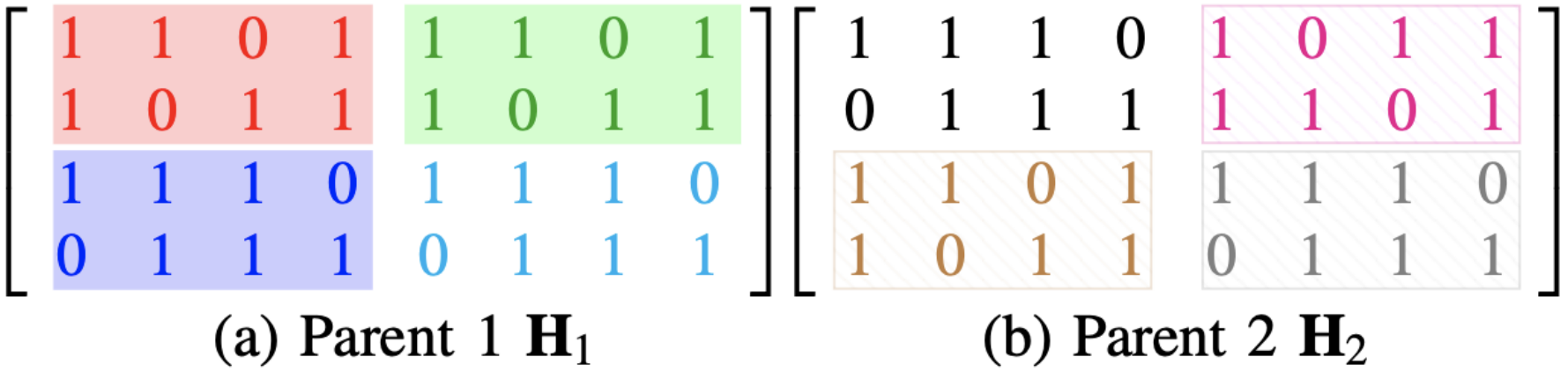
Как это применялось у нас

- Начальная популяция: случаный набор LDPC матриц
- Отбор на основе данных после декодирования
- Мутация – случайное добавление или уничтожение ребра в Таннеровом графе
- Скрещивание – кроссовер
- Когда придём к тому, что хотели – выберем лучший код из итоговой популяции

Как это применялось у нас

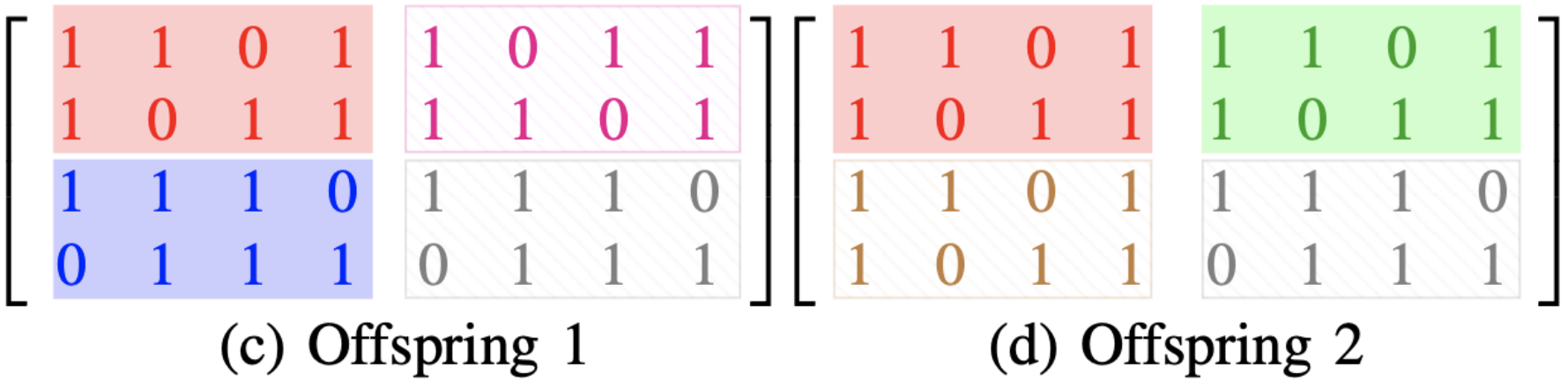


Как выглядят кроссоверы



(a) Parent 1 $\mathbf{H}_1$

(b) Parent 2 $\mathbf{H}_2$



(c) Offspring 1

(d) Offspring 2

Fig. 4: Crossover examples between the two parents $\mathbf{H}_1$ and $\mathbf{H}_2$.

Что ещё можно сделать?

- Более интересные мутации или кроссовер

Что ещё можно сделать?

- Более интересные мутации или кроссовер
- Добавить других нелинейных функций

Что ещё можно сделать?

- Более интересные мутации или кроссовер
- Добавить других нелинейных функций
- Нормально сравнить с известными кодами

Что ещё можно сделать?

- Более интересные мутации или кроссовер
- Добавить других нелинейных функций
- Нормально сравнить с известными кодами
- Найти бааагг