

УДК 621.391 : 519.725

© 2023 г. П.В. Трифонов

ПОСТРОЕНИЕ И ДЕКОДИРОВАНИЕ ПОЛЯРНЫХ КОДОВ С БОЛЬШИМИ ЯДРАМИ: ОБЗОР¹

Представлены методы построения и декодирования полярных кодов с большими ядрами. Важнейшей проблемой при реализации алгоритма последовательного исключения декодирования полярных кодов и его обобщений является обработка ядра, т.е. быстрое вычисление логарифмических отношений правдоподобия для входных символов ядра. Представлены оконный и рекурсивный решетчатый методы обработки больших ядер. Рассмотрены методы оценки надежности битовых подканалов и получения кодов с улучшенными свойствами расстояния.

Ключевые слова: полярные коды, большие ядра.

DOI: 10.31857/S0555292323010035, **EDN:** RMFRTN

§ 1. Введение

Полярные коды, предложенные Э. Ариканом [1], уже вошли в стандарт 5G. Однако их применение там ограничено каналом управления, где передаются относительно короткие блоки данных. Для больших блоков данных коды с малой плотностью проверок на четность (LDPC) оказываются более предпочтительны. Причиной этого является плохое масштабирование полярных кодов Арикана и их аналогов. Конструкцию полярных кодов можно обобщить, заменив матрицу Арикана размера 2×2 на большую [2]. Можно показать, что такие коды, известные как полярные коды с большими ядрами, достигают асимптотически оптимальной экспоненты масштабирования [3–5]. Такие коды могут быть декодированы с помощью алгоритма последовательного исключения (ПИ) и его обобщений со сложностью $O(n \log n)$. Однако при непосредственной реализации константа, скрытая в асимптотической записи, растет экспоненциально с размером ядра ℓ . Поэтому до недавнего времени такие коды считались непрактичными. Построение полярных кодов с большими ядрами может быть выполнено методами, основанными на тех же принципах, что и в случае полярных кодов Арикана.

В данном обзоре представлен обзор методов построения и декодирования полярных кодов с большими ядрами.

§ 2. Полярные коды

2.1. Поляризация канала. Рассмотрим вначале случай двоичных кодов. Полярный код [2] – множество векторов $c_0^{n-1} = u_0^{n-1} K^{\otimes m}$, где K – обратимая $(\ell \times \ell)$ -матрица, называемая ядром поляризации, $n = \ell^m$, $u_i = 0$ для $i \in \mathcal{F}$, $\mathcal{F} \subset [n]$ называется множеством замораживания, $u_a^b = (u_a, u_{a+1}, \dots, u_b)$ и $[n] = \{0, \dots, n-1\}$. Это

¹ Работа выполнена при финансовой поддержке Российского научного фонда в рамках гранта РНФ № 22-11-00208.

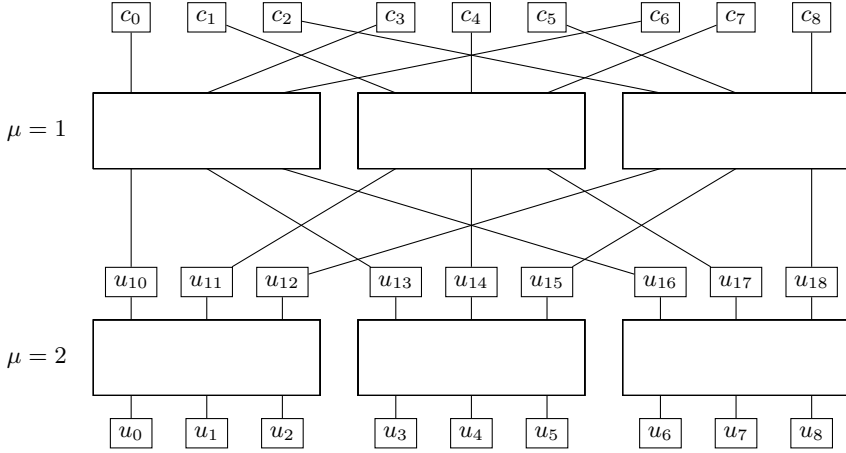


Рис. 1. Схема кодирования для полярного кода длины 9 на основе ядра размерности 3

определение может быть обобщено на случай кодов со смешанными ядрами [6, 7], кодовые слова которых равны $c_0^{n-1} = u_0^{n-1}(K_{\ell_0} \otimes K_{\ell_2} \otimes \dots \otimes K_{\ell_{m-1}})$, где K_{ℓ_i} – ядро размерности ℓ_i . Если не указано иное, мы рассматриваем случай, когда все ядра одинаковы.

На рис. 1 приведен пример схемы кодирования для полярного кода с ядром размерности $\ell = 3$ и $m = 2$ слоями поляризующего преобразования.

Можно показать, что матрица $K^{\otimes m}$ вместе с симметричным каналом без памяти с двоичным входом и функцией переходных вероятностей $W(r|c)$ задают синтетические битовые подканалы с функциями переходных вероятностей

$$W_m^{(i)}(r_0^{n-1}, u_0^{i-1} | u_i) = \frac{1}{2^{n-1}} \sum_{u_{i+1}^{n-1}} \prod_{j=0}^{n-1} W(r_j | (u_0^{n-1} K^{\otimes m})_j), \quad 0 \leq i < \ell^m,$$

где r_j – выходные символы канала. Если K нельзя преобразовать перестановками столбцов в верхнетреугольную матрицу, то пропускные способности этих битовых подканалов сходятся к 0 или 1, а доля подканалов с пропускной способностью, близкой к 1, сходится к $I(W)$ – симметричной пропускной способности исходного канала W (см. [2]). Множество замораживания $\mathcal{F}$ обычно выбирается как набор индексов i , соответствующих подканалам с малой пропускной способностью $W_m^{(i)}$. Удобно определить вспомогательные вероятности

$$W_m^{(i)}(u_0^i | r_0^{n-1}) = \frac{W_m^{(i)}(r_0^{n-1}, u_0^{i-1} | u_i)}{2W_m(r_0^{n-1})}.$$

Заметим, что иногда полярные коды определяются [1] с матрицей перестановки P в схеме кодирования, т.е. $c_0^{n-1} = u_0^{n-1} P K^{\otimes m}$. Такая перестановка нужна только для удобства обозначений и может быть исключена без изменения свойств полученного кода. Следует лишь учесть ее наличие или отсутствие в декодере.

2.2. Алгоритм последовательного исключения декодирования полярных кодов.

Рассмотрим передачу кодового слова c_0^{n-1} полярного кода по симметричному каналу без памяти. Декодирование полярных кодов может быть реализовано алгоритмом

последовательного исключения (ПИ), который принимает решения

$$\hat{u}_i = \begin{cases} 0, & i \in \mathcal{F}, \\ \arg \max_{u_i \in \mathbb{F}_2} W_m^{(i)}(\hat{u}_0^{i-1} \bullet u_i | r_0^{n-1}), & i \notin \mathcal{F}, \end{cases}$$

где $\bullet$ обозначает оператор конкатенации,

$$W_m^{(i)}(u_0^i | r_0^{n-1}) = A \sum_{u_{i+1}^{n-1}} \prod_{j=0}^{n-1} W_0^{(0)}((u_0^{n-1} K^{\otimes \ell})_i | r_i),$$

$$W(c | r) = \frac{W(r | c)}{2W(r)},$$

а A – нормирующий множитель, не зависящий от u_i . Из рекурсивной структуры произведения Кронекера следует, что эти вероятности могут быть вычислены как

$$W_\mu^{(\ell i+s)}(u_0^{li+s} | r_0^{\ell \mu -1}) = A' \sum_{u_{i+s+1}^{\ell i+\ell-1}} \prod_{j=0}^{\ell-1} W_{\mu-1}^{(i)}((u_{\ell t}^{\ell(t+1)-1} K)_j, t \in [i+1] | r_{j,\ell}^{\ell \mu -1}), \quad (1)$$

где A' – еще один нормирующий множитель, $r_{j,\ell}^{n-1} = (r_j, r_{j+\ell}, \dots, r_{j+n-\ell})$, $0 \leq s < \ell$, $0 \leq i < \ell \mu -1$, $1 \leq \mu \leq m$. Эта операция известна как обработка ядра или маргинализация ядра [8, 9]. Непосредственная реализация этой операции имеет сложность $O(\ell 2^\ell)$. Более эффективные методы обсуждаются в § 4.

Известно, что алгоритм ПИ является крайне субоптимальным. Для случая кодов с ядром Арикана было предложено множество его усовершенствований [10–13], среди которых наиболее известным является списочный алгоритм последовательного исключения Талья – Варди. Большинство этих методов допускает обобщение на случай неарикановских ядер. Для этого необходимо внести следующие изменения:

- Реализовать соответствующий алгоритм обработки ядра;
- Если алгоритм обработки ядра требует сохранения некоторой информации о состоянии на различных фазах s , обеспечить доступ к этой информации при рассмотрении декодером различных путей $u_0^{\ell i+s}$.

2.3. Параметры ядра. Корректирующая способность полярных кодов конечной длины в значительной степени зависит от вероятности битовых ошибок в подканалах, используемых для передачи незамороженных символов. Она может быть оценена сверху параметром Бхаттачарьи $Z(W_m^{(i)})$ этих подканалов. Следующий параметр [2] показывает, насколько хорошими могут быть подканалы, полученные из матрицы поляризующего преобразования $K^{\otimes m}$.

Определение 1. Матрица K имеет *скорость поляризации* $E(K)$, если для любого канала с двоичным входом W , $0 \leq I(W) < 1$, выполнено следующее:

- Для любого $\beta < E(K)$ справедливо $\liminf_{m \rightarrow \infty} \mathbf{P}\{Z(W_m^{(i)}) \leq 2^{-\ell m \beta}\} = I(W)$;
- Для любого $\beta > E(K)$ справедливо $\liminf_{m \rightarrow \infty} \mathbf{P}\{Z(W_m^{(i)}) \geq 2^{-\ell m \beta}\} = 1$.

Здесь вероятность следует интерпретировать как долю подканалов $W_m^{(i)}$, удовлетворяющих указанным свойствам.

Из этого определения следует, что для любого R , $0 < R < I(W)$, для достаточно больших m вероятность ошибки декодирования методом последовательного исключения для полярного $(n = \ell^m, Rn)$ -кода ограничена сверху величиной 2^{-n^β} , $\beta < E(K)$.

Определение 2 (см. [2]). Частичные расстояния $\mathcal{D}_i$, $0 \leq i < \ell$, $(\ell \times \ell)$ -матрицы $K = \begin{pmatrix} K[0] \\ \dots \\ K[\ell-1] \end{pmatrix}$ определяются как

$$\mathcal{D}_i = d_H(K[i], \mathcal{C}^{(i+1)}), \quad 0 \leq i < \ell - 1, \quad (2)$$

$$\mathcal{D}_{\ell-1} = \text{wt}(K[\ell-1]), \quad 0 \leq i < \ell - 1, \quad (3)$$

где $\mathcal{C}^{(i)} = \langle K[i], \dots, K[\ell-1] \rangle$ – линейный $(\ell, \ell-i)$ -код, порожденный строками ядра $i, \dots, \ell-1$, а $d_H(a, C)$ – минимальное расстояние Хэмминга между вектором a и кодовыми словами кода C .

Вектор $\mathcal{D}_0^{\ell-1}$ называется профилем частичных расстояний. Скорость поляризации ядра K может быть вычислена как $E(K) = \frac{1}{\ell} \sum_{i=0}^{\ell-1} \log_{\ell} \mathcal{D}_i$. Можно легко проверить, что для случая ядра Арикана $K_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ частичными расстояниями являются $\mathcal{D}_0 = 1$, $\mathcal{D}_1 = 2$, так что скорость его поляризации $E(K_2) = 0,5$. Можно показать [2], что существуют $(\ell \times \ell)$ -ядра K_{ℓ} с $E(K_{\ell}) \rightarrow 1$.

Экспонента масштабирования μ для семейства кодов со скоростью R показывает длину $n = O\left(\frac{1}{(I(W) - R)^{\mu}}\right)$, необходимую для достижения некоторой фиксированной целевой вероятности ошибки на кодовое слово на канале W с симметричной пропускной способностью $I(W)$. Для случайных кодов можно показать [14], что $\mu = 2$. Для произвольных семейств кодов неизвестно, существует ли такая μ . Можно предположить, что для полярных кодов для любого ε существует $\mu(W, K_{\ell})$ и

$$f = \lim_{m \rightarrow \infty} \beta_m \ell^{\frac{m}{\mu(W, K_{\ell})} - m}, \quad 0 < f < \infty,$$

где β_m – число подканалов $W_m^{(i)}$, таких что $\varepsilon \leq Z(W_m^{(i)}) \leq 1 - \varepsilon$ (см. [15, 16]). Если это предположение верно, то можно показать, что полярные коды Арикана на двоичном стирающем канале имеют экспоненту масштабирования $\mu(BEC, K_2) = 3,627$. Точное значение экспоненты масштабирования для полярных кодов на других каналах неизвестно. Однако можно показать [15], что для любого симметричного канала без памяти с двоичным входом она удовлетворяет условию $3,579 \leq \mu(W, K_2) \leq 4,714$. Можно показать [3], что существуют $(\ell \times \ell)$ -ядра K_{ℓ} с $\lim_{\ell \rightarrow \infty} \mu(BEC, K_{\ell}) = 2$. Более того [17], существует ядро размера 64 с экспонентой масштабирования $\mu \approx 2,87$, что уже лучше, чем $\mu_{\text{SC-LDPC}} = 3$ – эвристическая оценка экспоненты масштабирования пространственно связанных LDPC-кодов [18]. В работе [5] было показано, что комбинируя несколько тщательно построенных локальных ядер, можно получить экспоненту масштабирования $\mu = 2 + \alpha$ для любого $\alpha > 0$ и любого симметричного канала без памяти с двоичным входом. Более того, для любого дискретного канала без памяти с пропускной способностью $I(W)$, любых $\pi, \rho > 0$, таких что $\pi + 2\rho < 1$, существуют полярноподобные коды длины n с вероятностью ошибки на кодовое слово $e^{-n^{\pi}}$, скоростью $R = I(W) - n^{-\rho}$ и сложностью декодирования $O(n \log n)$ (см. [4]).

2.4. Недвоичные полярные коды. Полярные коды могут быть построены для каналов с q -ичными входными алфавитами, $q > 2$. Однако если поляризующее преобразование построено недостаточно тщательно, может возникнуть многоуровневая поляризация, т.е. пропускные способности подканалов могут сходить к значениям, отличным от 0 или 1, q -ичных символов за одно использование канала, или поляризация может вообще не наблюдаться [19, 20]. В работе [21] было показано,

что обратимая $(\ell \times \ell)$ -матрица K над $GF(p^m)$ является поляризирующей, если для любого $\overline{K} = VKP$ расширение поля $GF(p)$, порожденное присоединением всех элементов $\overline{K}$ к $GF(p)$, равно $GF(p^m)$, где V – верхнетреугольная матрица, P – матрица перестановок, а $\overline{K}$ – нижнетреугольная матрица.

§ 3. Конструкции поляризирующих ядер

Задача нахождения поляризирующего ядра состоит из двух подзадач:

1. Нахождение допустимого профиля частичных расстояний (ПЧР) $\mathcal{D}_0^{\ell-1}$;
2. Нахождение конкретного ядра с заданным ПЧР.

Считается, что ПЧР допустим, если существует ядро с таким ПЧР. Для получения необходимых условий допустимости ПЧР можно использовать классические верхние границы минимального расстояния линейных блочных кодов [2, 22]. К сожалению, эти ограничения применимы только для случая монотонных ПЧР, т.е. $\mathcal{D}_0 \leq \mathcal{D}_1 \leq \dots \leq \mathcal{D}_{\ell-1}$. В [23, 24] было показано, что ядра с немонотонным ПЧР допускают гораздо более простую обработку по сравнению с ядрами с монотонным ПЧР, имеющими ту же скорость поляризации. Поиск условий допустимости для немонотонных ПЧР остается открытой проблемой.

Даже для фиксированного ПЧР нахождение удовлетворяющего ему ядра в общем случае остается трудной задачей [25]. Из-за большого пространства поиска необходимо использовать некоторые методы для быстрого определения неподходящих и эквивалентных кандидатов. В частности, в [24] было предложено использовать таблицы весов лидеров смежных классов и инварианты эквивалентности кодов для сокращения пространства поиска и получения ядер с почти оптимальной скоростью поляризации.

Простым способом получения большого ядра является использование некоторого семейства вложенных алгебраических кодов [2, 22]. Вложенные алгебраические коды были использованы в [17] совместно с компьютерным поиском для получения ядра размера 64 с экспонентой масштабирования $\mu \approx 2.87$. Имея некоторое поляризирующее ядро, можно попытаться переставить его столбцы, чтобы уменьшить сложность различных алгоритмов обработки ядра [26, 27]. Имея хорошее поляризирующее ядро, можно укоротить его, чтобы получить семейство меньших ядер, допускающих унифицированную реализацию обработчика [28].

§ 4. Обработка ядра

Важнейшей проблемой при реализации алгоритма ПИ является эффективное вычисление вероятностей (1). В [29] было предложено аппроксимировать эти значения² как

$$W_m^{(i)}(u_0^i | r_0^{n-1}) \approx \mathcal{W}_m^{(i)}(u_0^i | r_0^{n-1}) = \max_{u_{i+1}^{n-1}} \prod_{j=0}^{n-1} W_0^{(0)}\left((u_0^{n-1} K^{\otimes \ell})_j | r_j\right),$$

так что

$$\mathcal{W}_m^{(\ell i+s)}(u_0^{\ell i+s} | r_0^{n-1}) = \max_{u_{s+1}^{\ell-1}} \prod_{j=0}^{\ell-1} \mathcal{W}_{m-1}^{(i)}\left((u_{\ell t}^{\ell(t+1)-1} K)_j, t \in [i+1] | r_{j,\ell}^{n-1}\right), \quad (4)$$

и $\mathcal{W}_0^{(0)}(c|r) = W(c|r)$. Отметим, что $\mathcal{W}_m^{(i)}(u_0^i | r_0^{n-1})$ – вероятность наиболее вероятного продолжения вектора u_0^i , не учитывающая никаких ограничений замораживания на символы u_j , $i < j < n$. В [29] было показано, что эта аппроксимация

² Здесь мы опускаем коэффициенты нормировки, так как они не влияют на декодирование.

обеспечивает существенное снижение средней сложности последовательного декодирования с незначительной потерей корректирующей способности.

Определим логарифмические отношения правдоподобия (ЛОПП)

$$S_m^{(i)}(u_0^{i-1}, r_0^{n-1}) = \ln \frac{\mathcal{W}_m^{(i)}(u_0^{i-1} \bullet 0 | r_0^{n-1})}{\mathcal{W}_m^{(i)}(u_0^{i-1} \bullet 1 | r_0^{n-1})}.$$

Далее для простоты ограничимся рассмотрением случая $m = 1$. Предполагая, что все $u_j, i < j < \ell$, принимают значения 0 и 1 с вероятностью $1/2$, можно заметить, что вычисление $\mathcal{W}_1^{(i)}(u_0^i | r_0^{\ell-1})$ эквивалентно декодированию по максимуму правдоподобия $r_0^{\ell-1}$ в смежном классе линейного кода, порожденного строками $i + 1, \dots, \ell - 1$ ядра K , где представитель смежного класса задается линейной комбинацией с коэффициентами $u_0, \dots, u_i$ верхних $i + 1$ строк K . Пусть $\hat{c}_0^{\ell-1}$ – жесткие решения, соответствующие $r_0^{\ell-1}$.

Видно, что

$$\begin{aligned} S_1^{(i)}(u_0^{i-1}, r_0^{\ell-1}) &= \ln \frac{\max_{u_{i+1}^{\ell-1}} \prod_{j=0}^{\ell-1} W\left(\left((u_0^{i-1}, 0, u_{i+1}^{\ell-1})K\right)_j | r_j\right)}{\max_{u_{i+1}^{\ell-1}} \prod_{j=0}^{\ell-1} W\left(\left((u_0^{i-1}, 1, u_{i+1}^{\ell-1})K\right)_j | r_j\right)} = \\ &= \min_{u_{i+1}^{\ell-1}} \left(\sum_{j=0}^{\ell-1} \left(\log W(\hat{c}_j | r_j) - \log W\left(\left((u_0^{i-1}, 1, u_{i+1}^{\ell-1})K\right)_j | r_j\right) \right) \right) - \\ &- \min_{u_{i+1}^{\ell-1}} \left(\sum_{j=0}^{\ell-1} \left(\log W(\hat{c}_j | r_j) - \log W\left(\left((u_0^{i-1}, 0, u_{i+1}^{\ell-1})K\right)_j | r_j\right) \right) \right) = \\ &= \min_{u_{i+1}^{\ell-1}} M\left(\left(u_0^{i-1}, 1, u_{i+1}^{\ell-1}\right)K, S_0^{\ell-1}\right) - \min_{u_{i+1}^{\ell-1}} M\left(\left(u_0^{i-1}, 0, u_{i+1}^{\ell-1}\right)K, S_0^{\ell-1}\right) = \\ &= \frac{1}{2} \left(\max_{u_{i+1}^{\ell-1}} T\left(\left(u_0^{i-1}, 0, u_{i+1}^{\ell-1}\right)K, S_0^{\ell-1}\right) - \max_{u_{i+1}^{\ell-1}} T\left(\left(u_0^{i-1}, 1, u_{i+1}^{\ell-1}\right)K, S_0^{\ell-1}\right) \right), \end{aligned} \quad (5)$$

где $M(c_0^{\ell-1}, S_0^{\ell-1}) = \sum_{j: (-1)^{c_j} S_j < 0} |S_j|$ – корреляционная невязка вектора $c_0^{\ell-1}$ относительно вектора ЛОПП $S_0^{\ell-1}$, $T(c_0^{\ell-1}, S_0^{\ell-1}) = \sum_{j=0}^{\ell-1} (-1)^{c_j} S_j$ – корреляционная функция, а $S_j = \ln \frac{W(0 | r_j)}{W(1 | r_j)}$ – входные ЛОПП.

В симметричном канале имеем

$$M\left(\left(u_0^i, u_{i+1}^{\ell-1}\right)K, S_0^{\ell-1}\right) = M\left(u_{i+1}^{\ell-1}K[i+1 : \ell-1], \bar{S}_0^{\ell-1}\right),$$

где $\bar{S}_i = (-1)^{f_i} S_i$, $0 \leq i < \ell$, $f_i = u_0^i K[0 : i]$, а $K[a : b]$ обозначает подматрицу матрицы K , состоящую из строк $a, a + 1, \dots, b$. Это позволяет переформулировать задачу вычисления выражения (5) следующим образом. Пусть $\bar{K}^{(i)}$ – матрица, полученная добавлением к $K[i : \ell-1]$ столбца $(1, 0, \dots, 0)^T$. Пространство строк $\bar{K}^{(i)}$ называется i -м расширенным ядерным кодом $\bar{C}^{(i)}$. Этот код имеет длину $\ell + 1$ и размерность $\ell - i$. Предполагая, что $\bar{S}_\ell = 0$, получаем, что (5) может быть вычислено путем нахождения двух наиболее вероятных кодовых слов i -го расширенного ядерного кода, соответствующих $\bar{S}_0^\ell$, имеющих 0 и 1 в последнем символе. Это может быть реа-

лизовано с помощью алгоритма декодирования по максимуму правдоподобия для расширенного ядерного кода $\bar{C}^{(i)}$.

Примером такого алгоритма является алгоритм Витерби.

4.1. Применение рекурсивных решеток для обработки больших поляризующих ядер. Укажем следующие два подхода.

Рекурсивное декодирование по максимуму правдоподобия. Решетки позволяют эффективно реализовать декодирование линейных блочковых кодов по максимуму правдоподобия. Однако широко известный алгоритм Витерби не является оптимальным с точки зрения сложности. Существенное снижение сложности может быть получено при использовании рекурсивного алгоритма декодирования по максимуму правдоподобия [30]. Идея этого алгоритма заключается в рекурсивном разбиении полученного зашумленного вектора на несколько секций $[x, y)$, определении для каждой секции нескольких наиболее вероятных векторов $c_x^{y-1} \in \mathbb{F}_2^{y-x}$, соответствующих полученным значениям r_x^{y-1} , и их объединении для получения наиболее вероятных векторов для более длинных секций.

Пусть дан линейный блочковый код C , а также его подкод $C_{h,h'}$, такой что все его кодовые слова имеют ненулевые символы только в позициях $h \leq i < h'$. Пусть $p_{h,h'}(C)$ – линейный код, полученный путем выкалывания из кодовых слов C всех символов, кроме тех, которые находятся в позициях $h \leq i < h'$. Далее определим $s_{h,h'}(C) = p_{h,h'}(C_{h,h'})$, т.е. код, полученный из C путем его укорочения на все символы, кроме тех, которые имеют индексы $h \leq i < h'$. Коды $s_{h,h'}(C)$ и $p_{h,h'}(C)$ называются *секционными кодами*. Рассмотрим минимальную решетку кода C и ее секции, соответствующие символам от x до y . Можно показать [30], что пути между двумя соседними состояниями в этой секции соответствуют смежным классам в $p_{x,y}(C)/s_{x,y}(C)$. Эти смежные классы могут появляться в решетке несколько раз. Следовательно, можно упростить декодирование по максимуму правдоподобия (МП), предварительно вычислив метрики этих путей.

Более конкретно, для каждого смежного класса $D \in p_{x,y}(C)/s_{x,y}(C)$ необходимо определить наиболее вероятный элемент $\ell(D)$, т.е. элемент с минимальной корреляционной невязкой

$$M(D) = M(\ell(D), S_x^{y-1}),$$

где S_i , $x \leq i < u$, – ЛОПП, соответствующие декодируемому вектору. Пусть *таблица составных метрик* (ТСМ) $T_{x,y}$ – массив, содержащий значения $T_{x,y}[v].\ell = \ell(D)$ и $T_{x,y}[v].m = M(D)$, где v – номер смежного класса D . В случае обычного декодирования (n, k) -кода $p_{0,n}(C)/s_{0,n}(C)$ содержит единственный элемент, поэтому соответствующая ТСМ имеет одну запись $T_{0,n}[0]$, что дает решение задачи МП-декодирования.

Непосредственный подход к построению ТСМ для некоторого кода C заключается в переборе всех кодовых слов из $p_{x,y}(C)$, и нахождении наиболее вероятного из них для каждого смежного класса в $p_{x,y}(C)/s_{x,y}(C)$. Мы предполагаем, что этот метод используется для $y - x < 2$. Однако в [30] для случая $y - x \geq 2$ был предложен более эффективный подход. Пусть z таково, что $x < z < y$. Предположим, что порождающая матрица $p_{x,y}(C)$ представлена в виде

$$G_{x,y}^{(p)} = \begin{pmatrix} G_{x,z}^{(s)} & 0 \\ 0 & G_{z,y}^{(s)} \\ G_{x,y}^{(00)} & G_{x,y}^{(01)} \\ G_{x,y}^{(10)} & G_{x,y}^{(11)} \end{pmatrix}, \quad (6)$$

где $G_{x,y}^{(s)} = \begin{pmatrix} G_{x,z}^{(s)} & 0 \\ 0 & G_{z,y}^{(s)} \\ G_{x,y}^{(00)} & G_{x,y}^{(01)} \end{pmatrix}$ – порождающая матрица $s_{x,y}(C)$, а $G_{x,y}^{(00)}$ и $G_{x,y}^{(01)}$ – некоторые матрицы размера $k'_{x,y} \times (z-x)$ и $k''_{x,y} \times (y-z)$ соответственно, где $k'_{x,y} = k'_{x,y}(C)$ и $k''_{x,y} = k''_{x,y}(C)$ – некоторые целые числа, зависящие от кода. Эти числа и матрицы могут быть получены из минимальной спэновой (minimum-span) формы порождающей матрицы кода C .

Существует взаимно-однозначное соответствие между векторами вида $vG'_{x,y}$, где $G'_{x,y} = \begin{pmatrix} G_{x,y}^{(10)} & G_{x,y}^{(11)} \end{pmatrix}$ – матрица размера $k'_{x,y} \times (y-x)$, и смежными классами $D \in p_{x,y}(C)/s_{x,y}(C)$. В дальнейшем v используется в качестве номера смежного класса.

Видно, что

$$T_{x,y}[v].m = \min_{c_x^{y-1} \in D} M(c_x^{y-1}, r_x^{y-1}) = \min_{w \in \mathbb{F}_2^{k''_{x,y}}} (T_{x,z}[a].m + T_{z,y}[b].m), \quad v \in \mathbb{F}_2^{k'_{x,y}}, \quad (7)$$

где a и b – индексы смежных классов $D' \in p_{x,z}(C)/s_{x,z}(C)$ и $D'' \in p_{z,y}(C)/s_{z,y}(C)$ соответственно, таких что $(w \ v) \begin{pmatrix} G_{x,y}^{(00)} \\ G_{x,y}^{(10)} \end{pmatrix} \in D'$ и $(w \ v) \begin{pmatrix} G_{x,y}^{(01)} \\ G_{x,y}^{(11)} \end{pmatrix} \in D''$. Такие векторы a, b могут быть определены из системы уравнений

$$\begin{aligned} (a' \ a) \begin{pmatrix} G_{x,z}^{(s)} \\ G_{x,z}' \end{pmatrix} &= (w \ v) \begin{pmatrix} G_{x,y}^{(00)} \\ G_{x,y}^{(10)} \end{pmatrix}, \\ (b' \ b) \begin{pmatrix} G_{z,y}^{(s)} \\ G_{z,y}' \end{pmatrix} &= (w \ v) \begin{pmatrix} G_{x,y}^{(01)} \\ G_{x,y}^{(11)} \end{pmatrix}, \end{aligned}$$

где a', b' – векторы, необходимые для обеспечения совместности этой системы, но не используемые далее. Решение этой системы имеет вид $a = (w \ v)\hat{G}_{x,y}$, $b = (w \ v)\tilde{G}_{x,y}$ для некоторых матриц $\hat{G}_{x,y}$ и $\tilde{G}_{x,y}$. Соответствующий наиболее вероятный представитель смежного класса равен $T_{x,y}[v].\ell = T_{x,z}[\hat{a}].\ell \bullet T_{z,y}[\hat{b}].\ell$, где $\hat{a}, \hat{b}$ – значения a и b , дающие минимум в (7).

Сложность вычислений по такой схеме составляет $O(2^{k'_{x,y} + k''_{x,y}})$. Она может быть дополнительно снижена за счет использования приемов, предложенных в [30]. Общая сложность декодирования сильно зависит от используемого метода секционирования, т.е. правила выбора точки разбиения z для различных x, y . Этот подход, известный как рекурсивное декодирование по максимуму правдоподобия (РДМП), является более эффективным по сравнению с алгоритмом Витерби [30].

Обработка ядра. Значения (5) могут быть найдены путем применения алгоритма РДМП к смежным классам расширенных ядерных кодов. Пусть $\bar{\mathcal{C}}(u_0^{i-1}) = w + \bar{\mathcal{C}}^{(i)}$ – смежный класс $\bar{\mathcal{C}}^{(i)}$, задаваемый ранее принятыми решениями u_0^{i-1} , где $w = (u_0^{i-1}K[0 : i-1], 0)$. Для любой секции $[x, y]$ смежные классы, связанные с состояниями в рекурсивной решетке для $\bar{\mathcal{C}}(u_0^{i-1})$, получаются из смежных классов для $\bar{\mathcal{C}}^{(i)}$ как $D(u_0^{i-1}) = \{c + w_x^{y-1} \mid c \in D\}$, $D \in p_{x,y}(\bar{\mathcal{C}}^{(i)})/s_{x,y}(\bar{\mathcal{C}}^{(i)})$.

Поскольку представители смежных классов $\ell(D)$ не нужны в контексте задачи обработки ядра, в дальнейшем предполагается, что записи ТСМ содержат только значения m . Поэтому мы используем запись $T_{x,y}[v]$ вместо $T_{x,y}[v].m$. Также видно, что $p_{0,\ell}(\bar{\mathcal{C}}^{(i)})/s_{0,\ell}(\bar{\mathcal{C}}^{(i)})$ содержит два смежных класса (класса эквивалентности),

которые соответствуют $u_i = 0$ и $u_i = 1$. Следовательно,

$$S_1^{(i)}(u_0^{i-1}, r_0^{\ell-1}) = T_{0,\ell}[1] - T_{0,\ell}[0], \quad (8)$$

где $T_{0,\ell} - \text{TSM}$, построенная для $\bar{\mathcal{C}}(u_0^{i-1})$ и заданного зашумленного вектора $r_0^{\ell-1}$.

Более того, мы предлагаем повторно использовать TSM или их части, полученные на последовательных фазах i . Для этого нам необходимо определить, как TSM эволюционируют с i , найти способ обработки предыдущих решений u_0^{i-1} , разработать эффективные алгоритмы построения TSM для коротких секций и найти оптимальную стратегию секционирования.

Предположим, что для всех фаз i используется одно и то же секционирование. Очевидно, что $p_{x,y}(\bar{\mathcal{C}}^{(i+1)}) \subset p_{x,y}(\bar{\mathcal{C}}^{(i)})$ и $s_{x,y}(\bar{\mathcal{C}}^{(i+1)}) \subset s_{x,y}(\bar{\mathcal{C}}^{(i)})$, $i \in [\ell-1]$, для любых x, y , таких что $0 \leq x < y \leq \ell$.

Лемма 1. Если $p_{x,y}(\bar{\mathcal{C}}^{(i+1)}) = p_{x,y}(\bar{\mathcal{C}}^{(i)})$ и $s_{x,y}(\bar{\mathcal{C}}^{(i+1)}) = s_{x,y}(\bar{\mathcal{C}}^{(i)})$, то для любого $u_i \in \mathbb{F}_2$ таблица составных метрик $T_{x,y}$, построенная для $\bar{\mathcal{C}}(u_0^{i-1})$, идентична TSM $T'_{x,y}$, построенной для $\bar{\mathcal{C}}(u_0^i)$ для того же принятого вектора $r_0^{\ell-1}$.

Из приведенной выше леммы следует, что не нужно пересчитывать TSM для тех секций, где секционные коды не меняются от фазы i к фазе $i+1$.

В дальнейшем мы предполагаем, что

$$s_{x,z}(\mathcal{C}^{(i+1)}) = s_{x,z}(\mathcal{C}^{(i)}) \quad \text{и} \quad s_{z,y}(\mathcal{C}^{(i+1)}) = s_{z,y}(\mathcal{C}^{(i)}). \quad (9)$$

Будем также временно считать, что $u_0^{i-1} = 0$.

Даже если секционные коды меняются, все равно можно переиспользовать некоторые результаты, полученные на предыдущих фазах. Пусть $k'_{i,x,y} = k'_{x,y}(\mathcal{C}^{(i)})$ и $k''_{i,x,y} = k''_{x,y}(\mathcal{C}^{(i)})$. Во-первых, заметим, что если $k''_{i+1,x,y} = k''_{i,x,y}$, но $k'_{i+1,x,y} < k'_{i,x,y}$, то $p_{x,y}(\mathcal{C}^{(i+1)})/s_{x,y}(\mathcal{C}^{(i+1)}) \subset p_{x,y}(\mathcal{C}^{(i)})/s_{x,y}(\mathcal{C}^{(i)})$, так что соответствующая TSM на фазе $i+1$ может быть получена как подвектор TSM на фазе i .

Во-вторых, минимизацию можно производить рекурсивно и сохранять все промежуточные результаты. Этот подход был первоначально представлен в [23]. Более конкретно, переишем (7) как

$$T_{x,y}[v] = \min_{w \in \mathbb{F}_2^{k''_{i,x,y}}} (T_{x,z}[a] + T_{z,y}[b]) = \min_{w_{k''_{i,x,y}-1}} \dots \min_{w_1} \min_{w_0} (T_{x,z}[a] + T_{z,y}[b]). \quad (10)$$

Вместо того чтобы хранить в TSM конечные результаты минимизации в (10), можно хранить промежуточные результаты минимизации для всех w . Эти значения могут быть представлены в виде двоичного дерева для каждого $v \in \mathbb{F}_2^{k'_{x,y}}$, так что путь от корня в этом дереве задается значениями $w_{k''_{i,x,y}-1}, w_{k''_{i,x,y}-2}, \dots, w_0$. Под лесом минимизации будем понимать множество таких деревьев, полученных на некоторой фазе для данной секции. Поддеревья внутри леса могут быть проиндексированы переменными w, v .

Описанное выше дерево минимизации, построенное на некоторой фазе i_0 , может быть использовано для получения TSM для всех $i \geq i_0$, где выполняется (9). Пусть $i_1 > i_0$ – наименьшее целое число, при котором это не выполняется.

Лемма 2. Пусть $G''_{i,x,y} = \begin{pmatrix} G_{x,y}^{(00)} & G_{x,y}^{(01)} \end{pmatrix}$ и $G'_{i,x,y} = \begin{pmatrix} G_{x,y}^{(10)} & G_{x,y}^{(11)} \end{pmatrix}$ – матрицы, полученные из (6) для кода $\mathcal{C}^{(i)}$. Если все матрицы $G''_{i,x,y}$ являются вложенными, так что $G''_{i+1,x,y}$ занимает верхние строки $G''_{i,x,y}$ для любого i , такого что $i_0 \leq i < i_1$, то лес минимизации для фазы i , $i_0 \leq i \leq i_1$, можно получить, взяв поддеревья, задаваемые векторами w и v , деревьев в лесу, построенном на фазе i_0 ,

$$(w \ v) \begin{pmatrix} G''_{i_0,x,y} \\ G''_{i_0,x,y} \end{pmatrix} = (\bar{w} \ \bar{v}) \begin{pmatrix} G''_{i,x,y} \\ G''_{i,x,y} \end{pmatrix}, \quad (11)$$

а $\bar{w} \in \mathbb{F}_2^{k''_{i_0,x,y}}$, $\bar{v} \in \mathbb{F}_2^{k''_{i,x,y}}$ обозначают индексы поддеревьев в лесу на фазе i .

Матрицу $G''_{i,x,y}$ всегда можно привести к форме, требуемой леммой 2, с помощью элементарных операций над строками.

Декодер ПИ должен учитывать на фазе i значения u_0^{i-1} . Это сводится к декодированию в смежных классах секционных кодов. Соответствующий представитель смежного класса для секции $[x, y]$ может быть вычислен как линейная комбинация подвекторов ядра $(K_{j,x} \ K_{j,x+1} \ \dots \ K_{j,y-1})$, $0 \leq j < i$.

Рассмотрим сначала случай секции $[x, y]$, где лес минимизации (10) строится заново. Если существует решение

$$(f_{j,x,y} \ h_{j,x,y}) \begin{pmatrix} G_{i,x,y}^{(s)} \\ G'_{i,x,y} \end{pmatrix} = (K_{j,x} \ K_{j,x+1} \ \dots \ K_{j,y-1}), \quad (12)$$

то тогда ТСМ для секции $[x, y]$ содержит в позиции $h_{j,x,y}$ запись, соответствующую требуемому представителю смежного класса. В этом случае полагаем $h_{j,x',y'} = 0$ для всех $x < x' < y' < y$. В противном случае будем считать, что $h_{j,x,y} = 0$. Имея вектор ранее принятых решений u_0^{i-1} , соответствующее смещение позиции на секции $[x, y]$

может быть вычислено как $h_{x,y} = \sum_{j=0}^{i-1} u_j h_{j,x,y}$, так что (10) преобразуется в

$$T_{x,y}[v] = \min_w (T_{x,z}[a + h_{x,z}] + T_{z,y}[b + h_{z,y}]). \quad (13)$$

Значения $h_{x,y}$ аналогичны частичным суммам, которые возникают в декодере ПИ полярных кодов Арикана.

Для тех секций, где ТСМ получается путем взятия поддеревьев (10), необходимо проверить, существует ли решение

$$\omega_{j,x,y} \begin{pmatrix} G_{i_0,x,y}^{(s)} \\ G'_{i_0,x,y} \end{pmatrix} = (K_{j,x} \ K_{j,x+1} \ \dots \ K_{j,y-1}). \quad (14)$$

Если это уравнение не имеет решения для некоторого $j < i$, то соответствующие представители смежных классов уже учтены на меньших секциях при построении ТСМ для секции $[x, y]$ на фазе i_0 , и можно считать, что $\omega_{j,x,y} = 0$. В противном случае, соответствующая ТСМ может быть получена из леса минимизации, построенного на фазе $i_0 < i$, как записи с номерами

$$(w \ v) = (\bar{w} \ \bar{v}) M_{i,x,y} + \sum_{j=0}^{i-1} u_j \omega_{j,x,y}.$$

Видно, что результат (8) не меняется, если из всех записей ТСМ на любом участке вычесть одно и то же значение. Это позволяет построить некоторые ТСМ с меньшей сложностью по сравнению с (13). Можно выделить следующие особые случаи:

1. Рассмотрим случай $k'_{i,x,y} = k''_{i,x,y} = 1$, $\hat{G}_{i,x,y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\tilde{G}_{i,x,y} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Это соответствует построению ТСМ с записями

$$T = [\min(w_{00} + w_{01}, w_{10} + w_{11}), \min(w_{10} + w_{01}, w_{00} + w_{11})],$$

где

$$w_{ij} = \begin{cases} 0, & i = \hat{c}_j, \\ |S_j|, & i \neq \hat{c}_j, \end{cases}$$

а $\hat{c}_j$ – жесткое решение, соответствующее ЛОПП S_j . Здесь S_j может быть либо канальным ЛОПП, либо величиной $T'[1] - T'[0]$, где T' – TCM из двух элементов, соответствующая левой или правой подсекции секции $[x, y]$.

Вычтем из обоих элементов T значение $w_{10} + w_{11}$. Видно, что в результате получается $\tilde{T} = [\min(-S_0 - S_1, 0), \min(-S_1, -S_0)]$. Далее вычтем из обоих элементов $\tilde{T}$ значение $\min(-S_0 - S_1, 0)$. В результате получим

$$\hat{T} = [0, \min(-S_1, -S_0) - \min(-S_0 - S_1, 0)] = [0, \text{sgn}(S_0) \text{sgn}(S_1) \min(|S_0|, |S_1|)].$$

Этот прием позволяет построить TCM для секции, используя только одну операцию сравнения, вместо двух сравнений и четырех сложений для непосредственной реализации. Отметим, что $\hat{T}[1]$ может быть также использован как S_j в аналогичной схеме в большей секции.

2. $k'_{i,x,y} = 1$, $k''_{i,x,y} = 0$, $\hat{G}_{i,x,y} = \tilde{G}_{i,x,y} = (1)$, что соответствует построению TCM с записями $T_{x,y} = [T_{x,z}[h_{x,z}] + T_{z,y}[h_{z,y}], T_{x,z}[h_{x,z} + 1] + T_{z,y}[h_{z,y} + 1]]$. Это можно преобразовать к

$$\hat{T}_{x,y} = [0, (-1)^{h_{x,z}}(T_{x,z}[1] - T_{x,z}[0]) + (-1)^{h_{z,y}}(T_{z,y}[1] - T_{z,y}[0])].$$

Если можно гарантировать, что $T_{x,z}[0] = T_{z,y}[0] = 0$, то $\hat{T}_{x,y}$ может быть построен с помощью всего одного сложения. Если $z - x = 1$ или $y - z = 1$, будем считать, что $T_{x,z} = [0, S_x]$ или $T_{z,y} = [0, S_z]$ соответственно, где S_j – соответствующий ЛОПП.

3. $k'_{i,x,y} = 0$, $k''_{i,x,y} = 2$, $\hat{G}_{i,x,y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\tilde{G}_{i,x,y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Пусть $a = T_{x,z}[1] - T_{x,z}[0]$, $b = T_{z,y}[1] - T_{z,y}[0]$. Эти значения можно получить без вычислений, если выполняется $T_{x,z}[0] = T_{z,y}[0] = 0$. Пусть $\hat{a}$ и $\hat{b}$ – жесткие решения, соответствующие a и b . Тогда получаем

$$\begin{aligned} T_{x,y}[\hat{a}, \hat{b}] &= 0, & T_{x,y}[1 \oplus \hat{a}, 1 \oplus \hat{b}] &= |a| + |b|, \\ T_{x,y}[1 \oplus \hat{a}, \hat{b}] &= |a|, & T_{x,y}[\hat{a}, 1 \oplus \hat{b}] &= |b|. \end{aligned}$$

Это можно вычислить всего за одну операцию.

Можно легко проверить, что в случае $K = BK_2^{\otimes m}$, где B – матрица перестановки обращения битов, и равномерного секционирования (т.е. для всех x, y используется $z = (x + y)/2$) TCM для всех секций могут быть построены с помощью описанных выше приемов 1 и 2. Более того, применение этих приемов в процессе работы алгоритма ПИ в точности повторяет min-sum версию алгоритма ПИ для полярных кодов Арикана [31].

Общая сложность обработки ядра предложенным алгоритмом задается величиной

$$C = \sum_{i=0}^{\ell-1} (\delta_i + c_{i,0,\ell}),$$

где δ_i – сложность вычисления конечного ЛОПП из полученной TCM, а $c_{i,x,y}$ – сложность построения TCM для участка $[x, y]$ на фазе i . В общем случае первая операция сводится к вычислению (8), т.е. требует одного вычитания. Однако если

на участке $[0, \ell)$ возникают описанные выше особые случаи 1 или 2, то $\delta_i = 0$. Более того,

$$c_{i,x,y} = \begin{cases} m_{ixy}, & \text{если ТСМ для подсекций переиспользуются,} \\ m_{ixy} + c_{i,x,z} + c_{i,z,y} & \text{в противном случае,} \end{cases}$$

где

$$m_{ixy} = \begin{cases} 0, & \text{если переиспользование леса возможно,} \\ 1, & \text{если встречается особый случай,} \\ 2^{k'_{ixy} + k''_{ixy}} + 2^{k'_{ixy}}(2^{k''_{ixy}} - 1) & \text{в противном случае.} \end{cases}$$

Первый и второй члены в последнем выражении представляют собой количество сложений и сравнений соответственно. Для каждой секции $[x, y)$ необходимо найти позицию разбиения z , которая минимизирует общую сложность. Это можно сделать заранее с помощью алгоритма оптимизации, приведенного в [30]. Отметим, что секционирование должно быть оптимизировано совместно для всех фаз i . Этот алгоритм использует метод динамического программирования для нахождения оптимального z и соответствующей сложности $c_{x,y} = \sum_{i=0}^{\ell-1} c_{i,x,y}$ для каждой допустимой

комбинации x и y . Результаты, полученные для коротких секций, сохраняются и повторно используются для более длинных секций. Такая оптимизация может значительно снизить общую сложность обработки. Было показано, что данный подход обеспечивает существенное снижение сложности обработки по сравнению с реализацией на основе алгоритма Витерби [32].

Следует признать, что описанный выше подход не требует решения каких-либо систем уравнений во время декодирования. Он сводится к сложению и сравнению элементов некоторых массивов, где индексы аргументов этих операций получаются как XOR некоторых предварительно вычисленных значений и смещений, задаваемых частичными суммами решений алгоритма ПИ.

4.2. Оконный метод обработки. Ядро Арикана допускает исключительно простую обработку. Действительно, (1) и (4) в случае ядра Арикана включают не более двух членов. Можно попытаться использовать эти простые выражения для обработки ядер большей размерности с лучшими поляризующими свойствами. Рассмотрим для простоты случай ядра K размерности $\ell = 2^\mu$. Поскольку матрица K обратима, можно выразить выходной вектор ядра как $c_0^{\ell-1} = u_0^{\ell-1} K = f_0^{\ell-1} K_2^{\otimes \mu}$, где $u_0^{\ell-1} = f_0^{\ell-1} T$, а $T = K_2^{\otimes \mu} K^{-1}$ – матрица перехода. Действительно, векторы f_0^{q-1} и u_0^{q-1} удовлетворяют системе уравнений

$$\underbrace{(S \ I)}_{\Theta'} (u_{\ell-1} \ \dots \ u_1 \ u_0 \ f_0 \ f_1 \ \dots \ f_{\ell-1})^T = 0,$$

где $(\ell \times \ell)$ -матрица S получается транспонированием T^{-1} и записью столбцов в полученной матрице в обратном порядке. Применяя элементарные операции со строками, матрицу Θ' можно преобразовать к минимальной спэновой форме Θ , где i -я строка начинается в i -м столбце и заканчивается в столбце z_i , где все z_i различны, и $\Theta_{i,z_i} = 1$. Это позволяет получить ограничения динамического замораживания в виде

$$f_{\omega_i} = \sum_{s=0}^i u_s \Theta_{\ell-1-i, \ell-1-s} + \sum_{t=0}^{\omega_i-1} f_t \Theta_{\ell-1-i, \ell+t}, \quad (15)$$

где $\omega_i = z_{\ell-1-i} - \ell$. Следовательно,

$$W_1^{(i)}(u_0^i | r_0^{\ell-1}) = \sum_{f_0^{h_i} \in \mathcal{Z}(u_0^i)} \widetilde{W}_\mu^{(h_i)}(f_0^{h_i} | r_0^{\ell-1}), \quad (16)$$

где $\widetilde{W}_\mu^{(i)}$ – вероятности путей (1), полученные для случая ядра Арикана K_2 , суммирование производится по множеству $\mathcal{Z}(u_0^i)$ векторов $f_0^{h_i}$, удовлетворяющих (15), и

$$h_i = \max_{0 \leq i' \leq i} \omega_{i'}.$$

Вычисление (16) сводится к перебору возможных значений символов u_s , $s \in [h_i + 1] \setminus \{\omega_t | 0 \leq t \leq i\}$. Число таких символов равно $\delta_i = h_i - i$. Это число называется размером окна. Такой перебор может быть реализован с помощью алгоритма списочного декодирования Талья – Варди с размером списка $2^{\max_i \delta_i}$, так что операции удаления пути не выполняются.

Аналогично (4) можно заменить суммирование в (16) на максимизацию и преобразовать алгоритм в область ЛОПП. Это приводит к следующим выражениям для ЛОПП, задаваемых матрицей Арикана:

$$\widetilde{S}_\mu^{(2i)}(f_0^{2i-1}, r_0^{2\mu-1}) = \text{sgn}(a) \text{sgn}(b) \min(|a|, |b|), \quad (17)$$

$$\widetilde{S}_\mu^{(2i+1)}(f_0^{2i}, r_0^{2\mu-1}) = (-1)^{u_{2i}} a + b, \quad (18)$$

где $a = \widetilde{S}_{\mu-1}^{(i)}(f_{0,2}^{2i-1} + f_{1,2}^{2i-1}, r_{0,2}^{2\mu-1})$, $b = \widetilde{S}_{\mu-1}^{(i)}(f_{1,2}^{2i-1}, r_{1,2}^{2\mu-1})$, а $\widetilde{S}_0^{(0)}(r_i) = S_i$ – входные ЛОПП. Вес пути, рассматриваемого min-sum версией декодера Талья – Варди, определяется [33] как

$$R(f_0^i, r_0^{\ell-1}) = R(f_0^{-1}, r_0^{\ell-1}) + \tau(\widetilde{S}_\mu^{(i)}(f_0^{i-1}, r_0^{\ell-1}), f_i), \quad (19)$$

где $R(r_0^{\ell-1}) = 0$ и

$$\tau(S, c) = \begin{cases} 0, & (-1)^c S \geq 0, \\ |S|, & (-1)^c S < 0, \end{cases}$$

является штрафной функцией. Можно показать [34], что

$$R(f_0^i, r_0^{\ell-1}) = \rho - \ln \max_{f_{i+1}^{\ell-1}} \widetilde{W}_\mu^{(\ell-1)}(f_0^{\ell-1} | r_0^{\ell-1}),$$

где ρ не зависит от f_0^i . Кроме того, $R(f_0^{\ell-1}, r_0^{\ell-1}) = M(f_0^{\ell-1} K_2^{\otimes \mu}, S_0^{\ell-1})$. Это позволяет [35] вычислить ЛОПП для входных символов ядра как

$$S_1^{(i)}(u_0^{i-1}, r_0^{\ell-1}) = \min_{f_0^{h_i} \in \mathcal{Z}(u_0^{i-1}, 1)} R(f_0^{h_i}, r_0^{\ell-1}) - \min_{f_0^{h_i} \in \mathcal{Z}(u_0^{i-1}, 0)} R(f_0^{h_i}, r_0^{\ell-1}). \quad (20)$$

Сложность этого вычисления растёт экспоненциально по 2^{δ_i} . Однако возможны некоторые упрощения [35]:

1. Для некоторых i могут существовать более эффективные методы оценки величин $R(f_0^{h_i}, r_0^{\ell-1})$, $f_0^{h_i} \in \mathcal{Z}(u_0^{i-1}, b)$, $b \in \mathbb{F}_2$, по сравнению с (17)–(19). Например, быстрое преобразование Адамара может быть использовано, если незамороженные символы $s \in [h_i + 1] \setminus \{\omega_t | 0 \leq t \leq i\}$ задают код Риды – Маллера первого порядка или схожий с ним код.
2. Число различных значений ЛОПП $\widetilde{S}_\nu^{(j)}$, $0 < \nu < \mu$, может быть намного меньше, чем 2^{δ_i} – число путей, рассматриваемых декодером Талья – Варди. Причина

этого в том, что векторы частичных сумм $f_{0,2}^{2i-1} + f_{1,2}^{2i-1}$ и $f_{1,2}^{2i-1}$ на различных уровнях μ могут образовывать некоторое подпространство соответствующего линейного пространства. Эти значения могут быть предварительно вычислены и использованы для различных путей в декодере Таля – Варди.

3. Минимизация в (20) может быть выполнена с сохранением промежуточных результатов в виде дерева, которые могут быть повторно использованы на последующих фазах.

Можно целенаправленно строить ядра с заданной скоростью поляризации и достаточно малым размером окна [36]. Основной недостаток оконного алгоритма обработки заключается в том, что пока нет его обобщения на случай ядер размерности, отличной от 2^μ , а его сложность может быть довольно высокой, если ядро не построено должным образом.

4.3. Приближенные методы обработки ядра. В [29] было предложено использовать так называемый алгоритм декодирования box-and-match [37] для поиска кодовых слов кода $\bar{\mathcal{C}}^{(i)}$, минимизирующих корреляционную невязку в (5). Сложность этого метода может быть уменьшена за счет повторного использования результатов метода Гаусса на разных фазах. Более того, сортировка входных ЛОПП может быть выполнена однократно в каждом обработчике ядра.

В качестве альтернативы можно использовать описанный выше оконный алгоритм обработки с ограниченным размером списка, т.е. оставить только несколько векторов f_0^i с наибольшей оценкой $R(f_0^i, r_0^{\ell-1})$ (см. [8, 38]).

§ 5. Построение кодов

5.1. Оценка надежности битовых подканалов. Классические полярные коды предполагают, что множество замораживания $\mathcal{F}$ содержит индексы ненадежных битовых подканалов. Существует несколько подходов к оценке надежности битовых подканалов в случае ядра Арикана, в том числе моделирование методом Монте-Карло, рекурсивные выражения для двоичного стирающего канала [1], гауссовская аппроксимация [39], аппроксимация стохастически улучшенными/ухудшенными каналами [40], эволюция плотностей [41, 42]. Некоторые из этих методов допускают относительно простое обобщение на случай кодов с большими ядрами.

Метод Монте-Карло. Наиболее простым способом оценки вероятности ошибки в битовых подканалах является использование идеального декодера ПИ. Более конкретно, произвольным образом³ генерируется вектор u_0^{n-1} , вычисляется вектор $c_0^{n-1} = u_0^{n-1} K^{\otimes m}$, который передается по каналу, вычисляются $W_m^{(i)}(u_0^i | r_0^{n-1})$, $u_i \in \{0, 1\}$, $0 \leq i < n$, или $S_m^{(i)}(u_0^{i-1}, r_0^{n-1})$, и принимается решение $\hat{u}_i$ на основе полученных значений. Если $\hat{u}_i \neq u_i$, то значение i -го счетчика ошибок увеличивается. Необходимо отметить, что оценки $\hat{u}_i$ не используются на последующих фазах. Указанные действия выполняются достаточно большое число раз, после чего вычисляется частота ошибок в битовых подканалах. Недостатком этого метода является его высокая вычислительная сложность.

Построение кодов для двоичного стирающего канала. Если кодовые слова полярного кода передаются по двоичному стирающему каналу, то все $W_m^{(i)}$ также являются двоичными стирающими каналами. Это позволяет анализировать их поведение, изучая свойства только одного ядра.

Конфигурации стирания могут быть связаны с двоичными векторами e , такими, что $e_i = 1$, если стирается i -й символ. Конфигурация стирания $e \in \mathbb{F}_2^\ell$ на фазе i неисправима, если существуют векторы $u_{i+1}^{\ell-1}, v_{i+1}^{\ell-1}$, такие что для всех u_0^{i-1} и для всех j ,

³ Для симметричных каналов достаточно рассмотреть $u_0^{n-1} = 0$.

таких что $e_j = 0$, выполняется $((u_0^{i-1}, 0, u_{i+1}^{\ell-1})K)_j = ((u_0^{i-1}, 1, v_{i+1}^{\ell-1})K)_j$, т.е. выходные символы ядра невозможно различить на нестертых позициях. Пусть $E_{i,w}$ – число неисправимых конфигураций стирания с весом w на фазе i . Тогда вероятность стирания (т.е. параметр Бхаттачарьи) в $W_1^{(i)}$ может быть вычислена как

$$f_i(z) = \sum_{w=0}^{\ell} E_{i,w} z^w (1-z)^{\ell-w},$$

где z – вероятность стирания в исходном канале. Набор функций $f_0(z), \dots, f_{\ell-1}(z)$ называется поляризационным поведением ядра. Оно может быть вычислено из решеток кодов $C^{(i)}$, $0 \leq i < \ell$ (см. [17, 43]).

Получив поляризационное поведение ядра, можно вычислить параметры Бхаттачарьи подканалов, задаваемых m -слоиным поляризационным преобразованием, как

$$Z_{m,\ell i+s} = f_s(Z_{m-1,i}), \quad 0 \leq i < \ell^{m-1}, \quad 0 \leq s < \ell,$$

где $Z_{0,0}$ – параметр Бхаттачарьи исходного двоичного стирающего канала. Поляризационное поведение также может быть использовано для вычисления экспоненты масштабирования ядра K (см. [16]).

Пример 1. Рассмотрим случай ядра Арикана $K_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Для фазы 0 имеются следующие неисправимые конфигурации стирания: (10), (01), (11). Для фазы 1 существует единственная неисправимая конфигурация стирания – (11). Следовательно,

$$Z_{m,2i} = f_0(Z_{m-1,i}) = 2Z_{m-1,i}(1 - Z_{m-1,i}) + Z_{m-1,i}^2 = 2Z_{m-1,i} - Z_{m-1,i}^2, \quad (21)$$

$$Z_{m,2i+1} = f_1(Z_{m-1,i}) = Z_{m-1,i}^2. \quad (22)$$

Этот подход может быть легко обобщен на случай недвоичных полярных кодов.

5.2. Гауссовская аппроксимация. Симметричная пропускная способность i -го подканала, задаваемого ядром K , может быть вычислена [44] как

$$I_1^{(i)} = 1 - \int_{-\infty}^{\infty} p_i(\xi | 0) \log_2(1 + e^{-\xi}) d\xi, \quad (23)$$

где $p_i(\xi | 0)$ – функция плотности вероятности отношения правдоподобия $L_1^{(i)} = \ln \frac{W_1^{(i)}(R_0^{\ell-1}, 0 | 0)}{W_1^{(i)}(R_0^{\ell-1}, 0 | 1)}$ при условии того, что передаются нулевые символы, а $R_0^{\ell-1}$ – случайные величины, соответствующие выходу канала. Такой интеграл может быть вычислен методом Монте-Карло, т.е. путем подачи случайных векторов $R_0^{\ell-1}$ на блок вычисления ЛОПП и усреднения соответствующих значений.

Оказывается, что подстановка вместо $L_1^{(i)}$ приближительных ЛОПП, задаваемых выражением (5), приводит к недостоверным результатам (например, $I_1^{(i)} < 0$). Вместо этого можно аппроксимировать (23) как

$$\begin{aligned} I_1^{(i)} &\approx 1 - \int_{-\infty}^{\infty} f_i(\psi | 0) \log_2 \left(1 + \frac{\mathbf{P}\{u_i = 1 | \psi\}}{\mathbf{P}\{u_i = 0 | \psi\}} \right) d\psi = \\ &= 1 - \int_{-\infty}^{\infty} f_i(\psi | 0) \log_2 \left(1 + \frac{f_i(-\psi | 0)}{f_i(\psi | 0)} \right) d\psi, \end{aligned} \quad (24)$$

где $f_i(\psi|0)$ – функция плотности вероятности $S_1^{(i)}(\mathbf{0}|R_0^{\ell-1})$, а $\mathbf{P}\{u_i = c|\psi\}$ – вероятность события $u_i = c$ при условии $S_1^{(i)}(\mathbf{0}|R_0^{\ell-1}) = \psi$. Следовательно, можно оценить $I_1^{(i)}$ путем построения гистограммы для $f_i(\psi|0)$ на основе результатов работы алгоритма, вычисляющего (5).

Для построения полярного кода с некоторым ядром K можно предположить, что все подканалы $W_\lambda^{(i)}$, $0 \leq \lambda \leq m$, $0 \leq i < \ell^\lambda$, являются гауссовскими, так что они могут быть полностью охарактеризованы соответствующей взаимной информацией $I_\lambda^{(i)}$. В [45] было предложено построить таблицы $I_1^{(i)}(C)$ значений $I_1^{(i)}$ для некоторого конечного набора параметров АБГШ-канала W с двоичным входом, где C – пропускная способность канала W . Эти таблицы могут быть использованы для интерполяции значений $I_1^{(i)}(C)$ для любого $C \in [0, 1]$.

Для построения полярного (ℓ^m, k) -кода для АБГШ-канала с пропускной способностью C можно рекурсивно вычислить пропускные способности битовых подканалов

$$I_m^{(\ell i+j)}(C) \approx I_1^{(j)}(I_{m-1}^{(i)}(C)), \quad 0 \leq j < \ell, \quad m > 1, \quad (25)$$

где $I_0^{(0)}(C) = C$, и объявить замороженными символы u_i , соответствующие подканалам с наименьшим $I_m^{(i)}(C)$. Этот подход может быть обобщен на случай кодов с недвоичными ядрами [46].

5.3. Коды с улучшенными дистантными свойствами. Полярные коды с большими ядрами имеют достаточно малое минимальное расстояние. Вследствие этого их корректирующая способность даже при декодировании списочным алгоритмом ПИ оказывается недостаточной. Она может быть улучшена путем использования полярных подкодов, полярных кодов с CRC [10] или других каскадных конструкций.

В большинстве случаев наилучшая корректирующая способность достигается при использовании полярных подкодов. Их построение опирается на концепцию динамически замороженных символов. Вместо того чтобы задавать $u_{j_i} = 0$, $j_i \in \mathcal{F}$, $0 \leq i < n - k$, можно потребовать, чтобы замороженные символы удовлетворяли условиям

$$u_{j_i} = \sum_{s < j_i} V_{i,s} u_j, \quad j_i \in \mathcal{F}, \quad (26)$$

где V – матрица ограничений размера $(n - k) \times n$, такая что последние ненулевые элементы ее строк находятся в различных столбцах $j_i \in \mathcal{F}$. Символы u_{j_i} с хотя бы одним ненулевым членом в правой части (26) называются динамически замороженными (DFS), а символы с $V_{i,s} = 0$, $s < j_i$, – статически замороженными. Такие коды называются полярными подкодами [47]. Их декодирование может быть реализовано простым обобщением списочного алгоритма ПИ.

Один из способов построения полярного $(n = \ell^m, k, d)$ -подкода состоит в том, чтобы взять базовый $(n, k' > k, d)$ -код с проверочной матрицей H , построить для него матрицу ограничений $V' = QH(K^{\otimes m})^T$, где обратимая матрица Q выбирается так, чтобы последние ненулевые элементы строк V' располагались в различных столбцах $j_0, \dots, j_{n-k'-1}$, и получить $V = \begin{pmatrix} V' \\ V'' \end{pmatrix}$. Здесь матрица V'' содержит $k' - k$ строк веса 1, ненулевые элементы которых расположены в столбцах j_i , соответствующих наименее надежным битовым подканалам $W_m^{(j_i)}$, где $j_i \notin \{j_0, \dots, j_{n-k'-1}\}$, $n - k' \leq i < n - k$. В качестве базовых кодов в сочетании с расширенными ядрами БЧХ могут быть выбраны, например, расширенные коды БЧХ [47].

Лучшую корректирующую способность можно получить с помощью рандомизированной конструкции, предложенной в [48]. Эта конструкция позволяет также комбинировать различные ядра в поляризующем преобразовании. Эта конструкция опирается на следующее утверждение.

Теорема 1. *Рассмотрим полярный (n, k) -код $\mathcal{C}$, заданный поляризующим преобразованием $A = K_{\ell_0} \otimes \dots \otimes K_{\ell_{m-1}}$ и множеством замораживания $\mathcal{F}$, где K_{ℓ_i} – ядро размерности ℓ_i , частичные расстояния $\mathcal{D}_{i,j}$ которого удовлетворяют условиям*

$$\mathcal{D}_{i,j} = \text{wt}(K_{\ell_i}[j]), \quad 0 \leq j < \ell_i, \quad 0 \leq i < m, \quad (27)$$

$n = \prod_{i=0}^{m-1} \ell_i$, а $K_{\ell_i}[j]$ – j -я строка матрицы K_{ℓ_i} . Тогда:

1. Минимальное расстояние полярного кода равно $d = \min_{s \notin \mathcal{F}} \text{wt}(A[s])$, где $A[s]$ – s -я строка матрицы A , $0 \leq s < n$;
2. Любое кодовое слово $c_0^{n-1} = u_0^{n-1} A \in \mathcal{C}$ веса d , где d – минимальное расстояние кода $\mathcal{C}$, имеет $u_s = 1$ для некоторого s , такого что $\text{wt}(A_s) = d$.

Чтобы получить (n, k) -код с хорошей корректирующей способностью при декодировании списочным алгоритмом ПИ, необходимо исключить из полярного кода ненулевые кодовые слова с малым весом (НКСМВ). Это можно сделать, введя ограничения динамического замораживания (26). Чтобы получить (n, k, d) -код $\mathcal{C}$, можно построить сначала полярный $(n, k + f_A, \bar{d})$ -код $\bar{\mathcal{C}}$ (родительский код) с матрицей ограничений $\bar{V}$, и построить матрицу ограничений для кода $\mathcal{C}$ как $V = \begin{pmatrix} \bar{V} \\ V^{(A)} \end{pmatrix}$. Здесь $V^{(A)}$ – $(f_A \times n)$ -матрица, которая определяет такие ограничения динамического замораживания, что большинство НКСМВ родительского кода не удовлетворяют им, т.е. $\mathcal{C}$ не содержит этих кодовых слов. Определим $\bar{\mathcal{F}}$ как множество индексов замороженных символов, заданное матрицей $\bar{V}$.

Чтобы уменьшить вероятность того, что правильный путь будет исключен из рассмотрения списочным декодером ПИ на ранних фазах, эти ограничения замораживания должны быть наложены на символы u_{s_i} с наименьшими возможными индексами s_i таким образом, чтобы кодовые слова с малым весом были исключены. Из теоремы 1 следует, что это можно сделать, обобщив конструкцию [49], т.е. выбрав в качестве s_i f_A максимальных индексов, таких, что $\text{wt}(A_{s_i}) = \bar{d}$, где $\bar{d}$ – минимальное расстояние родительского полярного кода $\bar{\mathcal{C}}$, и выбрав $V_{i,j}^{(A)}$, $0 \leq j < s_i$, $0 \leq i < f_A$, в качестве независимых случайных двоичных величин. Кроме того, положим $V_{i,s_i}^{(A)} = 1$ и $V_{i,j}^{(A)} = 0$, $j > s_i$. Действительно, такой выбор s_i гарантирует, что для любого НКСМВ $c_0^{n-1} = u_0^{n-1} A \in \bar{\mathcal{C}}$ можно найти такие значения $V_{i,j}^{(A)}$, что u_0^{n-1} не удовлетворяет (26), т.е. $c_0^{n-1} \notin \mathcal{C}$. Для малых значений f_A может оказаться невозможным исключить таким образом все НКСМВ, но результаты моделирования показывают, что даже при случайном выборе $V^{(A)}$ полученные коды обеспечивают достаточно хорошую производительность по сравнению с полярными кодами с CRC и LDPC-кодами. Формируемые таким образом ограничения динамического замораживания называются ограничениями типа A.

Можно дополнительно усовершенствовать конструкцию полярных подкодов путем уменьшения вероятности того, что списочный декодер ПИ исключит правильный путь на ранних фазах.

Вероятность $\mathcal{E}(L)$ события, соответствующего тому, что вес (19) правильного пути станет больше, чем вес L неправильных путей на некоторой промежуточной фазе списочного алгоритма декодирования, зависит от того, насколько быстро веса неправильных путей расходятся с весом правильного пути. Поэтому можно ве-

сти дополнительный набор динамически замороженных символов, отображаемых на относительно надежные битовые подканалы, так, чтобы для неправильного пути оценки этих символов, получаемые из ЛОПП $S_m^{(i)}$, с высокой вероятностью отклонялись от значений, использованных кодером, что приводило бы к уменьшению веса соответствующего пути.

Пусть C_i – пропускная способность (или другая мера надежности) битового подканала $W_m^{(i)}$, и рассмотрим последовательность $r_i : C_{r_0} \leq C_{r_1} \leq \dots \leq C_{r_{n-1}}$. В [49] было предложено наложить нетривиальные ограничения динамического замораживания на f_B символов u_{r_i} , передаваемых по подканалам с индексами $r_{n-k-f_A-1}, \dots, r_{n-k-f_A-f_B}$, т.е. по наиболее надежным битовым подканалам, используемых для передачи статически замороженных символов в классическом полярном коде. Здесь f_B – параметр конструкции кода. Такие ограничения динамического замораживания называются ограничениями типа В.

Вместо случайного выбора коэффициентов $V_{i,s}$ в (26) можно пытаться явно найти такие значения, которые минимизируют число НКСМВ в полученном коде [48, 50].

Как правило, полярные (под)коды с большими ядрами требуют меньшего размера списка для достижения той же корректирующей способности, что и коды, основанные на ядре Арикана. Если использовать эффективные алгоритмы обработки ядра вместе с тщательно подобранными поляризующими ядрами, то количество арифметических операций, выполняемых списочным алгоритмом ПИ для случая кодов на основе больших ядер, может быть меньше по сравнению с кодами на основе ядра Арикана при одинаковой корректирующей способности [35, 51].

Другой способ улучшить корректирующую способность полярных кодов с большими ядрами при декодировании списочным алгоритмом ПИ заключается в использовании модифицированной процедуры выбора номеров замороженных символов, как описано в [7].

§ 6. Заключение

Известно, что полярные коды с большими ядрами обеспечивают асимптотически оптимальную экспоненту масштабирования и скорость поляризации. В данной статье был представлен обзор методов построения и декодирования таких кодов. Многие из методов построения кодов, разработанных для случая кодов с ядром Арикана, относительно просто обобщаются на случай кодов с большими ядрами.

Сложность непосредственной реализации декодера таких кодов крайне высока. Однако для тщательно построенных поляризующих ядер могут быть построены алгоритмы обработки с низкой сложностью. Это позволяет полярным (под)кодам, основанным на больших ядрах, достичь такой же корректирующей способности при декодировании списочным алгоритмом последовательного исключения, как и кодам, основанным на ядре Арикана, с меньшей сложностью декодирования. В зависимости от структуры ядра, наименьшую сложность обработки могут обеспечивать оконный алгоритм или алгоритм на основе рекурсивных решеток. Это позволяет предположить, что более эффективные методы обработки ядра могут быть получены путем комбинирования этих подходов.

СПИСОК ЛИТЕРАТУРЫ

1. *Arıkan E.* Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels // IEEE Trans. Inform. Theory. 2009. V. 55. № 7. P. 3051–3073. <https://doi.org/10.1109/TIT.2009.2021379>
2. *Korada S.B., Şaşıoğlu E., Urbanke R.* Polar Codes: Characterization of Exponent, Bounds, and Constructions // IEEE Trans. Inform. Theory. 2010. V. 56. № 12. P. 6253–6264. <https://doi.org/10.1109/TIT.2010.2080990>

3. *Fazeli A., Hassani H., Mondelli M., Vardy A.* Binary Linear Codes with Optimal Scaling: Polar Codes with Large Kernels // IEEE Trans. Inform. Theory. 2021. V. 67. № 9. P. 5693–5710. <https://doi.org/10.1109/TIT.2020.3038806>
4. *Wang H.-P., Duursma I.M.* Polar Codes' Simplicity, Random Codes' Durability // IEEE Trans. Inform. Theory. 2021. V. 67. № 3. P. 1478–1508. <https://doi.org/10.1109/TIT.2020.3041570>
5. *Guruswami V., Riazanov A., Ye M.* Arian Meets Shannon: Polar Codes with Near-Optimal Convergence to Channel Capacity // IEEE Trans. Inform. Theory. 2022. V. 68. № 5. P. 2877–2919. <https://doi.org/10.1109/TIT.2022.3146786>
6. *Presman N., Shapira O., Litsyn S.* Mixed-Kernels Constructions of Polar Codes // IEEE J. Select. Areas Commun. 2016. V. 34. № 2. P. 239–253. <https://doi.org/10.1109/JSAC.2015.2504278>
7. *Bioglio V., Gabry F., Land I., Belfiore J.-C.* Multi-Kernel Polar Codes: Concept and Design Principles // IEEE Trans. Commun. 2020. V. 68. № 9. P. 5350–5362. <https://doi.org/10.1109/TCOMM.2020.3006212>
8. *Trifonov P.* Binary Successive Cancellation Decoding of Polar Codes with Reed–Solomon Kernel // Proc. 2014 IEEE Int. Symp. on Information Theory (ISIT'2014). Honolulu, HI, USA. June 29–July 4, 2014. P. 2972–2976. <https://doi.org/10.1109/ISIT.2014.6875379>
9. *Bioglio V., Land I.* On the Marginalization of Polarizing Kernels // Proc. 2018 IEEE 10th Int. Symp. on Turbo Codes & Iterative Information Processing (ISTC'2018). Hong Kong, China. Dec. 3–7, 2018. P. 1–5. <https://doi.org/10.1109/ISTC.2018.8625378>
10. *Tal I., Vardy A.* List Decoding of Polar Codes // IEEE Trans. Inform. Theory. 2015. V. 61. № 5. P. 2213–2226. <https://doi.org/10.1109/TIT.2015.2410251>
11. *Miloslavskaya V., Trifonov P.* Sequential Decoding of Polar Codes // IEEE Commun. Lett. 2014. V. 18. № 7. P. 1127–1130. <https://doi.org/10.1109/LCOMM.2014.2323237>
12. *Chandesris L., Savin V., Declercq D.* Dynamic-SCFlip Decoding of Polar Codes // IEEE Trans. Commun. 2018. V. 66. № 6. P. 2333–2345. <https://doi.org/10.1109/TCOMM.2018.2793887>
13. *Trifonov P.* A Score Function for Sequential Decoding of Polar Codes // Proc. 2018 IEEE Int. Symp. on Information Theory (ISIT'2018). Vail, CO, USA. June 17–22, 2018. P. 1470–1474. <https://doi.org/10.1109/ISIT.2018.8437559>
14. *Polyanskiy Y., Poor H.V., Verdú S.* Channel Coding Rate in the Finite Blocklength Regime // IEEE Trans. Inform. Theory. 2010. V. 56. № 5. P. 2307–2359. <https://doi.org/10.1109/TIT.2010.2043769>
15. *Mondelli M., Hassani S.H., Urbanke R.L.* Unified Scaling of Polar Codes: Error Exponent, Scaling Exponent, Moderate Deviations, and Error Floors // IEEE Trans. Inform. Theory. 2016. V. 62. № 12. P. 6698–6712. <https://doi.org/10.1109/TIT.2016.2616117>
16. *Fazeli A., Vardy A.* On the Scaling Exponent of Binary Polarization Kernels // Proc. 52nd Annu. Allerton Conf. on Communication, Control, and Computing (Allerton'2014). Monticello, IL, USA. Sept. 30–Oct. 3, 2014. P. 797–804. <https://doi.org/10.1109/ALLERTON.2014.7028536>
17. *Yao H., Fazeli A., Vardy A.* Explicit Polar Codes with Small Scaling Exponent // Proc. 2019 IEEE Int. Symp. on Information Theory (ISIT'2019). Paris, France. July 7–12, 2019. P. 1757–1761. <https://doi.org/10.1109/ISIT.2019.8849741>
18. *Mondelli M., Hassani S.H., Urbanke R.L.* How to Achieve the Capacity of Asymmetric Channels // IEEE Trans. Inform. Theory. 2018. V. 64. № 5. P. 3371–3393. <https://doi.org/10.1109/TIT.2018.2789885>
19. *Park W., Barg A.* Polar Codes for q -ary Channels, $q = 2^r$ // IEEE Trans. Inform. Theory. 2013. V. 59. № 2. P. 955–969. <https://doi.org/10.1109/TIT.2012.2219035>
20. *Şaşoğlu E., Telatar E., Arikan E.* Polarization for Arbitrary Discrete Memoryless Channels // Proc. IEEE 2009 Information Theory Workshop (ITW'2009). Taormina, Italy. Oct. 11–16, 2009. P. 144–148. <https://doi.org/10.1109/ITW.2009.5351487>
21. *Mori R., Tanaka T.* Source and Channel Polarization over Finite Fields and Reed–Solomon Matrices // IEEE Trans. Inform. Theory. 2014. V. 60. № 5. P. 2720–2736. <https://doi.org/10.1109/TIT.2014.2312181>

22. Presman N., Shapira O., Litsyn S., Etzion T., Vardy A. Binary Polarization Kernels from Code Decompositions // IEEE Trans. Inform. Theory. 2015. V. 61. № 5. P. 2227–2239. <https://doi.org/10.1109/TIT.2015.2409257>
23. Trofimiuk G., Trifonov P. Efficient Decoding of Polar Codes with Some 16×16 Kernels // Proc. IEEE 2018 Information Theory Workshop (ITW'2018). Guangzhou, China. Nov. 25–29, 2018. P. 11–15. <https://doi.org/10.1109/ITW.2018.8613307>
24. Trofimiuk G. A Search Method for Large Polarization Kernels // Proc. 2021 IEEE Int. Symp. on Information Theory (ISIT'2021). Melbourne, Australia. July 12–20, 2021. P. 2084–2089. <https://doi.org/10.1109/ISIT45174.2021.9517729>
25. Lin H.-P., Lin S., Abdel-Ghaffar K.A.S. Linear and Nonlinear Binary Kernels of Polar Codes of Small Dimensions with Maximum Exponents // IEEE Trans. Inform. Theory. 2015. V. 61. № 10. P. 5253–5270. <https://doi.org/10.1109/TIT.2015.2469298>
26. Moskovskaya E., Trifonov P. Design of BCH Polarization Kernels with Reduced Processing Complexity // IEEE Commun. Lett. 2020. V. 24. № 7. P. 1383–1386. <https://doi.org/10.1109/LCOMM.2020.2984382>
27. Abbasi F., Viterbo E. Large Kernel Polar Codes with Efficient Window Decoding // IEEE Trans. Veh. Technol. 2020. V. 69. № 11. P. 14031–14036. <https://doi.org/10.1109/TVT.2020.3029305>
28. Trofimiuk G. Shortened Polarization Kernels // Proc. 2021 IEEE Globecom Workshops (GC Wkshps). Madrid, Spain. Dec. 7–11, 2021. P. 1–6. <https://doi.org/10.1109/GCWkshps52748.2021.9681982>
29. Miloslavskaya V., Trifonov P. Sequential Decoding of Polar Codes with Arbitrary Binary Kernel // Proc. IEEE 2014 Information Theory Workshop (ITW'2014). Hobart, TAS, Australia. Nov. 2–5, 2014. P. 376–380. <https://doi.org/10.1109/ITW.2014.6970857>
30. Fujiwara T., Yamamoto H., Kasami T., Lin S. A Trellis-Based Recursive Maximum-Likelihood Decoding Algorithm for Binary Linear Block Codes // IEEE Trans. Inform. Theory. 1998. V. 44. № 2. P. 714–729. <https://doi.org/10.1109/18.661515>
31. Trifonov P. Trellis-Based Decoding Techniques for Polar Codes with Large Kernels // Proc. IEEE 2019 Information Theory Workshop (ITW'2019). Visby, Sweden. Aug. 25–28, 2019. P. 249–253. <https://doi.org/10.1109/ITW44776.2019.8989386>
32. Trifonov P., Karakchieva L. Recursive Processing Algorithm for Low Complexity Decoding of Polar Codes with Large Kernels // IEEE Trans. Commun. Early access June 2023, <https://doi.org/10.1109/TCOMM.2023.3285773>
33. Balatsoukas-Stimming A., Bastani Parizi M., Burg A. LLR-Based Successive Cancellation List Decoding of Polar Codes // IEEE Trans. Signal Process. 2015. V. 63. № 19. P. 5165–5179. <https://doi.org/10.1109/TSP.2015.2439211>
34. Trofimiuk G., Iakuba N., Rets S., Ivanov K., Trifonov P. Fast Block Sequential Decoding of Polar Codes // IEEE Trans. Veh. Technol. 2020. V. 69. № 10. P. 10988–10999. <https://doi.org/10.1109/TVT.2020.3006369>
35. Trofimiuk G., Trifonov P. Window Processing of Binary Polarization Kernels // IEEE Trans. Commun. 2021. V. 69. № 7. P. 4294–4305. <https://doi.org/10.1109/TCOMM.2021.3072730>
36. Trofimiuk G., Trifonov P. Construction of Binary Polarization Kernels for Low Complexity Window Processing // Proc. IEEE 2019 Information Theory Workshop (ITW'2019). Visby, Sweden. Aug. 25–28, 2019. P. 115–119. <https://doi.org/10.1109/ITW44776.2019.8989344>
37. Valembois A., Fossorier M. Box and Match Techniques Applied to Soft-Decision Decoding // IEEE Trans. Inform. Theory. 2004. V. 50. № 5. P. 796–810. <https://doi.org/10.1109/TIT.2004.826644>
38. Gupta B., Yao H., Fazeli A., Vardy A. Polar List Decoding for Large Polarization Kernels // Proc. 2021 IEEE Globecom Workshops (GC Wkshps). Madrid, Spain. Dec. 7–11, 2021. P. 1–6. <https://doi.org/10.1109/GCWkshps52748.2021.9681935>
39. Trifonov P. Efficient Design and Decoding of Polar Codes // IEEE Trans. Commun. 2012. V. 60. № 11. P. 3221–3227. <https://doi.org/10.1109/TCOMM.2012.081512.110872>
40. Tal I., Vardy A. How to Construct Polar Codes // IEEE Trans. Inform. Theory. 2013. V. 59. № 10. P. 6562–6582. <https://doi.org/10.1109/TIT.2013.2272694>

41. *Mori R., Tanaka T.* Performance of Polar Codes with the Construction Using Density Evolution // IEEE Commun. Lett. 2009. V. 13. № 7. P. 519–521. <https://doi.org/10.1109/LCOMM.2009.090428>
42. *Kern D., Vorköper S., Kühn V.* A New Code Construction for Polar Codes Using Min-Sum Density // Proc. 2014 8th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC'2014). Bremen, Germany. Aug. 18–22, 2014. P. 228–232. <https://doi.org/10.1109/ISTC.2014.6955119>
43. *Miloslavskaya V., Trifonov P.* Design of Binary Polar Codes with Arbitrary Kernels // Proc. 2012 IEEE Information Theory Workshop (ITW'2012). Lausanne, Switzerland. Sept. 3–7, 2012. P. 119–123. <https://doi.org/10.1109/ITW.2012.6404639>
44. *Richardson T., Urbanke R.* Modern Coding Theory. Cambridge, UK: Cambridge Univ. Press, 2008.
45. *Trifonov P.* On Construction of Polar Subcodes with Large Kernels // Proc. 2019 IEEE Int. Symp. on Information Theory (ISIT'2019). Paris, France. July 7–12, 2019. P. 1932–1936. <https://doi.org/10.1109/ISIT.2019.8849672>
46. *Karakchieva L., Trifonov P.* An Approximate Method for Construction of Polar Codes with Kernels over $\mathbb{F}_2^t$ // IEEE Commun. Lett. 2020. V. 24. № 9. P. 1857–1860. <https://doi.org/10.1109/LCOMM.2020.2995257>
47. *Trifonov P., Miloslavskaya V.* Polar Subcodes // IEEE J. Select. Areas Commun. 2016. V. 34. № 2. P. 254–266. <https://doi.org/10.1109/JSAC.2015.2504269>
48. *Trifonov P.* Randomized Polar Subcodes with Optimized Error Coefficient // IEEE Trans. Commun. 2020. V. 68. № 11. P. 6714–6722. <https://doi.org/10.1109/TCOMM.2020.3018781>
49. *Trifonov P., Trofimiuk G.* A Randomized Construction of Polar Subcodes // Proc. 2017 IEEE Int. Symp. on Information Theory (ISIT'2017). Aachen, Germany. June 25–30, 2017. P. 1863–1867. <https://doi.org/10.1109/ISIT.2017.8006852>
50. *Miloslavskaya V., Vucetic B., Li Y., Park G., Park O.-S.* Recursive Design of Precoded Polar Codes For SCL Decoding // IEEE Trans. Commun. 2021. V. 69. № 12. P. 7945–7959. <https://doi.org/10.1109/TCOMM.2021.3111625>
51. *Trifonov P.* Recursive Trellis Processing of Large Polarization Kernels // Proc. 2021 IEEE Int. Symp. on Information Theory (ISIT'2021). Melbourne, Australia. July 12–20, 2021. P. 2090–2095. <https://doi.org/10.1109/ISIT45174.2021.9517783>

Трифонов Петр Владимирович
 Университет ИТМО, Санкт-Петербург
 pvtrifonov@itmo.ru

Поступила в редакцию
 29.12.2022
 После доработки
 22.02.2023
 Принята к публикации
 22.02.2023