Локально восстанавливаемые коды на графах

Валинуров Д.Ю.[1]

Локально восстанавливаемые коды (LRC коды) это линейные коды с представляющим большой интерес для приложений свойством, что каждый символ кодового слова можно восстановить по небольшому множеству других символов. Символы кода можно интерпретировать как серверы с некоторой информацией. Тогда становится естественным задание топологии этих серверов в виде графа, в котором для каждого сервера указаны сервера, по данным из которых можно однозначно восстановить его данные. В статье рассматриваются LRC коды для некоторых типов топологий и приводятся оценки на скорость таких кодов.

Ключевые слова: коды исправляющие ошибки, локально восстанавливаемые коды, коды на графах.

1. Введение

С распространением систем хранения информации появилась необходимость в восстановлении стёртой информации с затратой минимального количества ресурсов. Одним из направлений исследований стали LRC коды, в которых стёртый символ может восстанавливаться по небольшому количеству других символов. Если представить систему хранения информации как множество датацентров, объединённых в некоторую топологию, то естественной становится задача восстановления потерянной информации с учётом этой топологии. В статье представлены оценки на скорость кода в произвольной топологии, заданной графом, и конструкция кода с субпакетизацией. Рассмотрена достижимость построенным кодом оценок на скорость кода в простейших топологиях — цикле и торе.

Обозначим через $\mathbb{F}_q$ конечное поле из q элементов. Назовём (n, k) кодом над алфавитом A подмножество $C \subseteq A^n$ мощности $|C| = A^k$. Далее

[1] *Валинуров Денис Юрьевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: denis.valinurow@yandex.ru.

Valinurov Denis Yurevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

будем рассматривать *линейные* $[n, k]$ коды над конечным полем $\mathbb{F}_q$ — k-мерное линейное подпространство $\mathbb{F}_q^n$, где k является размерностью кода $\dim(C)$. *Скоростью* $(n,\ k)$ кода назовём величину $R = \frac{k}{n}$.

Определение 1. Минимальным расстоянием *кода C называется величина* $d = \min_{x,y \in C, x \neq y} h(x, y)$, *где* h — расстояние Хэмминга, *то есть количество компонент, в которых векторы не равны. Нетрудно показать, что для линейных кодов* $d = \min_{x \in C, x \neq 0} w(x)$, *где* $w(x)$ - *вес слова* x, *то есть количество ненулевых компонент.*

Из минимального расстояния получается такая характеристика кода, как корректирующая способность. Код с минимальным расстоянием d может исправить $d - 1$ стираний.

Для линейных кодов известна оценка Синглтона $d \leqslant n - k + 1$. Коды, на которых достигаются равенство в этой оценке, называются *разделимыми кодами с максимальным расстоянием* или *МДР кодами*. Наиболее известным примером МДР кодов являются коды Рида-Соломона [2].

Обозначим за $[n]$ множество $\{1, 2, ..., n\}$. Ограничением $C|_A$ кода C на множество координат $A \subseteq [n]$ назовём код, полученный из C удалением всех координат не из A.

Определение 2. *Говорим, что* $(n,\ k)$ *код обладает свойством* r*-локальности, если выполняется следующее: для любого* $i \in [n]$ *существует подмножество* $R_i \subseteq [n] \setminus i, |R_i| \leq r$ *такое, что ограничения множества* $C(i, a) = \{x \in C : x_i = a\}$ *на* R_i *имеют пустое пересечение для* $a \neq a'$, *то есть* $C|_{R_i}(i, a) \bigcap C|_{R_i}(i, a') = \varnothing$.

Линейный код с таким свойством называется LRC $[n,\ k,\ r]$ кодом (locally recoverable code). Множества R_i будем называть *локальностями*. Из определения видно, что i-ый символ не может принимать разные значения при одинаковых значениях символов из R_i. Поэтому можно говорить, что символ i однозначно восстанавливается по множеству R_i и является функцией символов $\alpha_1, \alpha_2, ..., \alpha_r \in R_i$. Поэтому локальные коды обладают важным свойством, что каждый символ можно восстановить по небольшому количеству других символов, что позволяет избежать передач больших объемов данных для восстановленя.

В статье [3] доказаны основные оценки на скорость R и минимальное расстояние d для LRC кода: $d \leqslant n - k - \lceil \frac{k}{r} \rceil + 2,\ R \leqslant \frac{r}{r+1}$. LRC код называется *оптимальным*, если он достигает равенства в оценке на минимальное расстояние: $d = n - k - \lceil \frac{k}{r} \rceil + 2$.

Обычно сервера связаны в сеть, называемую топологией и задаваемую некоторым графом. Обозначим $G(V, E)$ — граф с множеством вер-

шин V и множеством рёбер E. Для произвольного графа $G(V, E)$ и произвольной вершины v в этом графе обозначим

$$N(v) = \{x \colon x \in V, (v, x) \in E\}.$$

Если S — множество, то примем $N(S) = \bigcup\limits_{s \in S} N(s)$.

Определение 3. *Пусть G — произвольный граф без петель и кратных рёбер с занумерованным множеством вершин $V = \{1, \ldots, n\}$. Говорим, что $[n, k, r]$ LRC код C является кодом для топологии задаваемой графом G, если для каждого i выполнено $N(i) = R_i$, то есть соседние вершины в графе G являются одновременно локальностью для этой вершины в коде C (для неориентированных графов считаем, что ребро идёт в обе стороны).*

Коды над топологиями слабо изучены в литературе, но многие оценки на эти коды следуют из оценок для произвольных LRC кодов.

В статье [6] представлена торическая топология локальных кодов. Кодовым словом является матрица $m \times n$. Символы можно восстанавливать локально по строке и по столбцу, то есть если в строке происходит не более a стираний, то их можно восстановить используя только символы этой строки с помощью $[n, n - a]$ кода C_{row}. Аналогично, если в столбце происходит не более b стираний, то их можно восстановить локально с помощью $[m, m - b]$ кода C_{col}.

В статье [8] представлены конструкции кодов для произвольных графов. Для неориентированных графов находится максимальное паросочетание, и информация в вершинах на концах одного ребра этого паросочетания дублируется. Для ориентированных графов дублируется информация во всех вершинах каждого цикла.

Определение 4. *Говорим, что код C для топологии $G(V = \{1, \ldots, n\}, E)$ исправляет локально e стираний, если для $x = (x_1, \ldots, x_n) \in C$ и произвольного $U \subset V, |U| \leqslant e$ выполнено следущее условие: для любого $i \in U$ существует функция f_i^U такая что $x_i = f_i^U(\{x_j \colon j \in \bigcup\limits_{l \in U} N(l) \setminus U\})$.*

LRC коды с несколькими стираниями в литературе называются ME-LRC кодами, оценки на скорость и расстояние таких кодов изучаются в [7]. Также в литературе встречаются t-LRC коды, где для каждого символа существует t непересекающихся локальностей, по которым этот символ может быть независимо восстановлен. Ясно, что t-LRC коды также могут локально восстанавливать любые t стираний. Оценки на t-LRC коды были изучены в [4].

В дальнейшем помимо линейных кодов будем рассматривать коды над алфавитом $A = \mathbb{F}_q^w$. Каждый символ такого кода будем интерпретировать как *сервер*, хранящий вектор из w значений из $\mathbb{F}_q$. В литературе параметр w называется субпакетизацией. Обозначим общую длину такого кода как $N = nw$, размерность – $K = kw$. Такие коды конкатенацией векторов для каждого сервера могут быть сведены к линейным кодам. Если на множестве серверов такого кода задана локальность, то соответствующий r-локальный код будем называть $[n, k, r, w]$ кодом.

Определение 5. *Введём расстояние между вершинами графа:* $\mathrm{dist}(v_1, v_2)$*, как минимальное количество рёбер в пути между вершинами* v_1 *и* v_2*. Назовём* r*-локальностью вершины* v *в графе* G *с функцией расстояния* dist *множество* $N(v, r) = \{v' : \mathrm{dist}(v, v') \leq r\}$.

Таким образом, далее будем считать, что каждая вершина графа по умолчанию хранит символ из $\mathbb{F}_q$, если же задан параметр w, то хранит вектор из $\mathbb{F}_q^w$. А рёбра графа задают локальности, то есть информация в вершине может быть однозначно восстановлена по информации из смежных вершин.

2. Оценки на код для произвольных топологий

В этом разделе приведём утверждения, верные для локальных кодов над произвольными графами. Как следствие теоремы 2.1.1 из статьи [3] получим следующее утверждение.

Утверждение 1. *Пусть* $G(V, E)$ *— ориентированный граф,* C *— код для топологии* G*,* $S \subseteq V$ *— подмножество вершин* G *такое, что подграф, натянутый на вершины* $V \setminus S$*, является ациклическим. Тогда имеем неравенство на размерность кода:* $\dim C \leqslant |S|$.

Доказательство. Рассмотрим подграф, натянутый на вершины $V \setminus S$. Из ацикличности следует, что в подграфе существует вершина v без исходящих рёбер. То есть локальность этой вершины целиком лежит в S, поэтому данные вершины v однозначно восстанавливаются по данным вершин из множества S.

Подграф, натянутый на вершины $V \setminus (S \cup \{v\})$, также является ациклическим, поэтому далее действуем аналогично. В итоге получается, что все вершины из $V \setminus S$ однозначно восстанавливаются по вершинам S, откуда получаем $\dim C \leqslant |S|$. □

Определение 6. *Разбиением неориентированного графа* G *на непересекающиеся клики назовём разбиение множества вершин* V *на непе-*

ресекающиеся подмножества $A_1, \ldots, A_l$ *такие, что* $\bigcup\limits_{i\in[l]} A_i = V$ *и для каждого* i *подграф, натянутый на вершины* A_i*, является полным.*

Лемма 1. *Пусть* $G(V, E)$ *— неориентированный граф,* $A_1, \ldots, A_l$ *— его разиение на непересекающиеся клики. Тогда существует код* C *для топологии* G*, исправляющий локально* e *стираний, для которого выполнено* $\dim C \geqslant |V| - e \cdot l$

Доказательство. В клике любые две вершины соединены ребром, поэтому на вершинах клики A_i мы можем построить $[|A_i|, |A_i| - e]$ МДР код C_i, который будет исправлять любые e стираний. Таким образом, если e стираний попадают в одну клику, то мы можем локально восстановить все стирания, используя код C_i. Если e стираний попадают в разные клики, то мы можем в каждой клике локально восстановить попавшие в эту клику стирания с помощью соответствующего кода.

Объединим все проверочные соотношения кодов C_i и получим проверочные соотношения кода C. Для кода C будет выполнено $\dim C = \sum\limits_{i\in[l]} \dim C_i = \sum\limits_{i\in[l]} (|A_i| - e) = |V| - e \cdot l$ □

Определение 7. *Набором независимых подмножеств* I *множества вершин* V *графа* G *назовём множества* $A_1, \ldots, A_l \subseteq V$ *такие, что для любых* $i \neq j$ *и любых* $x \in A_i, y \in A_j : x$ *и* y *не соединены ребром. Мощностью набора назовём число* $|I| = \bigcup\limits_{i\in[l]} |A_i|$*. Множеством соседних вершин набора назовём множество* $N(I) = \bigcup\limits_{i\in[l]} N(A_i) \setminus \bigcup\limits_{i\in[l]} A_i$*. Расстоянием набора назовём число* $D(I) = \max\limits_{i\in[l]} |A_i|$*.*

Лемма 2. *Пусть* $G(V, E)$ *– неориентированный граф,* C *- код для топологии* G*, исправляющий локально* e *стираний. Рассмотрим* I *– набор независимых подмножеств графа* G *с* $D(I) \leqslant e$*. Тогда* $\dim C \leqslant |V| - |I|$*.*

Доказательство. Для всех i выполнено $(N(A_i) \setminus A_i) \subseteq (V \setminus \bigcup\limits_{i\in[l]} A_i)$. Поэтому из свойства восстановления e стираний по локальным соотношениям получается, что все вершины из $\bigcup\limits_{i\in[l]} A_i$ можно восстановить по вершинам из $V \setminus \bigcup\limits_{i\in[l]} A_i$, откуда следует искомое неравенство. □

Замечание 1. *Для более точной оценки в лемме 3 необходимо находить наборы подмножеств с максимальной суммой* $\sum\limits_{i=1}^{l} |A_i|$*. В случае* $e = 1$ *задача состоит в нахождении максимального независимого множества, что эквивалентно нахождению минимального покрытия вершин и является NP-полной задачей.*

Как обобщение теоремы 14 из [8] и как следствие теоремы 2.1.2 из [3] может быть получено следующее утверждение.

Утверждение 2. *Пусть C – $[n, k, d]$ код для топологии $G(V, E)$, исправляющий локально e стираний. I – набор независимых подмножеств $G(V, E)$ такой, что $|N(I)| \leqslant k - 1$ и $D(I) \leqslant e$. Тогда $d \leqslant n - k + 1 - |I|$.*

Доказательство. Пусть $R = (\bigcup\limits_{A \in I} A) \cup N(I) \cup R'$, где R' – множество $k - 1 - |N(I)|$ произвольных вершин из $V \setminus N(I)$. Рассмотрим код $C|_R$, свойство восстанавливать локально e стираний у него остаётся, так как можно подставить в выколотые координаты кода C нули. Отсюда по лемме 3 получаем, что $\dim C|_R \leqslant k - 1$. То есть $\dim C|_R \neq \dim C$ и код $C|_R$ содержит нулевой вектор, поэтому $d \leqslant n - |R| \leqslant n - k + 1 - |I|$. $\square$

3. Конструкция для графов с функцией расстояния

В этом разделе приведём конструкцию исправляющего локально e стираний $[n, k, r, w]$ кода в виде композиции МДР кодов для произвольного неориентированного графа с заданной функцией расстояния. Под стиранием понимаем стирание всех w символов из одной вершины.

Определение 8. *Граф $G(V, E)$ называется (вершинно)* транзитивным, *если для любых двух вершин $v_1, v_2 \in V$ существует сохраняющее смежность биективное отображение множества вершин на себя $f \colon V \to V$ такое, что $f(v_1) = v_2$.*

У транзитивного графа окрестности каждой вершины устроены одинаково, и поэтому $|N(v_1, s)| = |N(v_2, s)|$ для любых вершин v_1, v_2 и расстояний s. Приведём конструкцию для случая транзитивного графа $G(V = \{1, \ldots, n\}, E)$ и $T|w$, где $T = |N(1, \frac{r}{2})|$. Но конструкцию можно рассматривать и для произвольного графа, что будет показано далее.

Построение 1. *Будем строить код C длины $N = nw$. Обозначим $\omega = \frac{w}{T}$. Для каждой вершины разобьём множество из w символов, входящих в эту вершину, на T частей, в каждой из которых ω символов. Таким образом, кодовое слово нашего кода C будет состоять из nT частей (каждая по ω символов), которые мы проиндексируем множеством $[n] \times [T]$.*

Зафиксируем некоторый $[w, w - e\omega]$ МДР код $\mathfrak{C}$, исправляющий $e\omega$ стираний. Далее для каждого $i \in [n]$ выберем из всех вершин из $N(i, \frac{r}{2})$ по одной части, не выбранной ранее, и обозначим через $V_i \subseteq [nw]$ — соответсвующее выбранным частям подмножество символов.

Добавим к коду C проверочные соотношения кода $\mathfrak{C}$, построенного на выбранных символах, т.е. потребуем чтобы каждое слово $c \in C$ удовлетворяло условию $c|_{V_i} \in \mathfrak{C}$. Для каждого $i \in [n]$ будет добавлено $e\omega$ проверочных соотношений, и все эти проверочные соотношения будут линейно независимы, так как будут являться проверочными соотношениями МДР кода, построенного на непересекающихся множествах символов. Поэтому код C будет иметь размерность $K = n(w - e\omega)$.

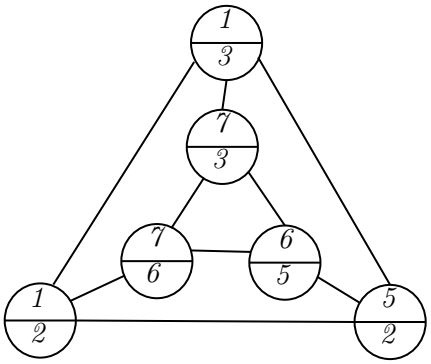
Теорема 1. *Код C из построения 1 является $[n, k, r, w]$ кодом для транзитивного графа $G(V, E)$ с заданной функцией расстояния, исправляющего локально e стираний, и его скорость $R = 1 - e/T$, где $T = |N(v, \frac{r}{2})|, v \in V$.*

Доказательство. Код $\mathfrak{C}$ имеет размерность $K = n(w - e\omega)$, откуда следует оценка на скорость $R = K/N = 1 - e/T$.

Нетрудно видеть, что вершины на расстоянии более $\frac{r}{2} + \frac{r}{2} = r$ не участвуют в одном проверочном соотношении, из чего следует r-локальность.

Выберем произвольные e вершин, без ограничения общности пусть это вершины $\{1, \ldots, e\}$, и докажем, что их можно восстановить по локальностям. Рассмотрим множество $S = \{V(i) \colon \exists v \in [e], i \in N(v, \frac{r}{2})\}$. Каждая часть любой из e стёртых вершин входит в некоторый $V(j) \in S$, поэтому эта часть может быть восстановлена по соответствующим проверочным соотношениям $C|_{V(j)}$, так как $\mathfrak{C}$ является $[w, w - e\omega]$ МДР кодом и может исправить стирание вплоть до e частей по ω символов в каждой. □

Замечание 2. *Конструкция лучше работает для чётных r, например, для $r = 1$ конструкция вообще не работает. В общем случае для построения V_i не обязательно выбирать вершины из множества $N(i, \frac{r}{2})$. Достаточно требовать, чтобы расстояние между любыми двумя выбранными вершинами не превосходило r. Например, для $r = 1$ нетрудно построить модифицированную конструкцию, которую покажем на следующем примере.*



Над этим графом строим код с параметрами $n = 6, r = 1, w = 2$. *Каждую вершина содержит два символоа, и в двух соседних вершинах два символа являются копией друг друга. Одинаковыми цифрами обозначены символы, которые являются копией друг друга. Скорость полученного кода равна* $1/2$.

Замечание 3. *Также конструкцию можно обобщить на случай нетранзитивных графов. Для таких графов графов будем делить множество символов в каждой вершине на разное количество частей, а именно вершину* i *поделим на* $|N(i, \frac{r}{2})|$ *частей. Части, на которые делится множество символов в вершине, также будем считать различными по размеру. Проиндексируем полученные части как ранее. Пусть множество* $K(i, j)$ *будет содержать те символы кода, которые входят в часть* j *вершины* $i \in [n]$, *где* $1 \le j \le |N(i, \frac{r}{2})|$.

Тогда для $K(i, j)$ *будут выполняться следующие равенства:* $\bigcup_{1 \le j \le |N(i, \frac{r}{2})|} K(i, j) = \{iw + 1, \dots, (i + 1)w\}$ *и* $K(i_1, j_1) \cap K(i_2, j_2) = \varnothing$ *для всех* $i, i_1, i_2 \in [n]$ *и* $(i_1, j_1) \ne (i_2, j_2)$. *То есть все части не пересекаются, и объединение всех частей одной вершины составляет в точности множество символов, содержащихся в этой вершине.*

В конструкции для нетранзитивных графов вместо одного МДР кода $\mathfrak{C}$ *будем рассматривать разные МДР коды* $\mathfrak{C}_i$, *каждый из которых будет построен на координатах* $V(i)$ *в обозначениях выше. Будем требовать, чтобы каждое слово* $c \in C$ *удовлетворяло условию* $c|_{V_i} \in \mathfrak{C}_i$.

Обозначим

$$n(i) = |V(i)|,$$

$$h(i) = \max_{\substack{(v_1, j_1), (v_2, j_2), \dots, (v_e, j_e), \\ \forall t \in [e]: \ K(v_t, j_t) \subset V(i)}} \sum_{t \in [e]} |K(v_t, j_t)|.$$

То есть величина $h(i)$ *равна максимальному размеру объединения произвольных* e *частей, входящих в* $V(i)$. *Тогда* $\mathfrak{C}_i$ *будет строиться как*

$[n(i), n(i)-h(i)]$ *МДР код на координатах* $V(i)$ *и будет исправлять стирание любого множества* e *входящих в него частей серверов, так как максимальный размер такого множества равен* $h(i)$*. Поэтому теорема 1 также будет верна, но для меньшего значения скорости, а именно* $R = \sum\limits_{i\in[n]} \frac{n(i)-h(i)}{nw}$.

Для уменьшения избыточности можно требовать, например, чтобы значение $h(i)/n(i)$ *было в точности равно* $e/|N(i, \frac{r}{2})|$*. В таком случае все части, входящие в* $V(i)$*, должны иметь одинаковый размер. Такое разбиение на части не всегда существует, что будет видно далее. Допустим, что для кода* $\mathfrak{C}_i$ *из каждой вершины* $v \in N(i, \frac{r}{2})$ *выбрали части с одинаковыми размерами* ω_i *(*ω_i *могут быть различны для* $i \in [n]$*). Тогда должно выполняться:* $\sum\limits_{j\in N(i,\frac{r}{2})} \omega_j = w$.

В итоге, для нахождения размерности кода получаем следующую задачу целочисленного линейного программирования, где максимизируемым значением является размерность искомого кода:

$$\begin{cases} nw - \sum\limits_{i\in[n]} \omega_i e \to \max; \\ \sum\limits_{j\in N(i,\frac{r}{2})} \omega_j = w, \ \ \forall i \in [n]; \\ \omega_i \geqslant 0, \ \ \forall i \in [n]. \end{cases}$$

4. Примеры LRC на графах

В этом разделе рассмотрим простейшие топологии: цикл и тор с заданными на них функциями расстояний, то есть локальностями вершины v будут множества $N(v, r)$. Для этих топологий построим коды с исправлением e стираний. Будем сравнивать параметры кодов для различных w с оценками для LRC с $w = 1$, что в общем случае является неверным, так как используя субпакетизацию также можно улучшать различные параметры. Построенная ранее конструкция достигает простейшую нижнюю оценку на скорость кода в рассматриваемых топологиях, но применима и для произвольного графа.

Определение 9. Предельной скоростью *семейства* $[n, k, r, w]$ *кодов над семейством графов* $\{G_n\}_{n=1}^{\infty}$ *назовём величину* $\tilde{R}(r, e, w) = \sup_{n\to\infty} R(G_n, e)$*, где* $R(G_n, e)$ *— наибольшая скорость* $[n, k, r, w]$ *кода над графом* G_n*, который может исправлять стирания* e *серверов.*
Предельной скоростью *кода конструкции 1 назовём величину* $\hat{R}(r, e, n) = \sup_{w\to\infty} R'(G_n, w, e)$*, где* $R'(G_n, w, e)$ *— скорость кода конструкции 1, исправляющего* e *стираний, над графом* G_n *с субпакетизацией* w.

4.1. Циклический граф

Циклическим графом Z_n назовём граф с n вершинами на рисунке 3. Зададим функцию расстояния и r-локальность согласно определению 5.

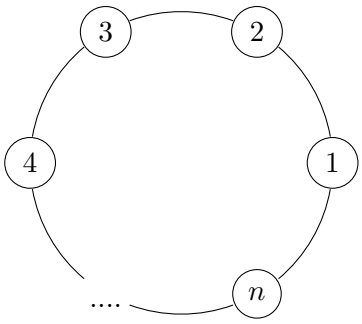


Рис. 1. Циклический граф локальности.

Для LRC с одним стиранием есть общая оценка на скорость кода из статьи [3]: $R \leqslant 1 - \frac{1}{2r+1}$, так как локальность в обычном смысле равна $2r$. Далее покажем, что оптимальная скорость для циклического графа меньше этого значения.

Утверждение 3. *Для циклического графа Z_n наибольшая скорость LRC кода, исправляющего e ошибок, удовлетворяет следующим неравенствам:*
$1 - e \times \lceil \frac{n}{r+1} \rceil / n \leqslant R(Z_n, e) \leqslant 1 - e \times \lfloor \frac{n}{r+e} \rfloor / n.$

Доказательство. Построим граф G, вершинами которого будут являться вершины Z_n, а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r.

Выберем в качестве набора независимых подмножеств графа G для леммы 2 множество $M = \{\{(r+e)i + j \colon 1 \leqslant j \leqslant e\} : 0 \leqslant i \leqslant \lfloor \frac{n}{r+e} \rfloor\}$. Тогда по лемме 2: $R \leqslant 1 - |M|/n = 1 - e \times \lfloor \frac{n}{r+e} \rfloor / n$.

Подряд идущие $r+1$ вершины графа G образуют клику, поэтому из леммы 1 получаем $R \geqslant 1 - e \times \lceil \frac{n}{r+1} \rceil / n$. □

Переходя в утверждении 3 к пределу при $n \to \infty$ получим следующую теорему.

Теорема 2. *Предельная скорость $[n, k, r, w]$ кода над семейством $\{Z_n\}$ удовлетворяет следующим неравенствам:* $1 - \frac{e}{r+1} \leqslant \tilde{R}(r, e, w) \leqslant 1 - \frac{e}{r+e}$.

Доказательство. Переход к $n \to \infty$ в утверждении 3. □

Естественно требовать, чтобы размер локальности каждой вершины не превышал общего количества вершин. Следующее утверждение верно для $2r + 1 \leqslant n$.

Утверждение 4. *Предельная скорость кода конструкции 1 для циклического графа равна* $\hat{R}(r,1,n) = \tilde{R}(r,1,w) = 1 - \frac{1}{r+1}$.

Доказательство. Следуя построению 1 и замечанию 2, для каждой вершины $i \in [n]$ выберем множество из $r+1$ последовательных вершин $\{i - \lfloor \frac{r}{2} \rfloor, \ldots, i + \lceil \frac{r}{2} \rceil\}$ (считается, что вершины зациклены). Если $r+1|w$, то разобьём выбранные вершины на части размера $\omega_i = \frac{w}{r+1}$, иначе по замечанию 3 размеры частей можно выбирать равными $\omega_i = \lfloor \frac{w}{r+1} \rfloor$ и $\omega_i = \lceil \frac{w}{r+1} \rceil$. Далее возьмём часть размера ω_i с каждой выбранной вершины и построим на этих частях проверочное соотношение. Возьмём все проверочные соотношения для каждого $i \in [n]$. Размерность итогового кода $K = nw - \lceil \frac{w}{r+1} \rceil n$, откуда $R = 1 - \lceil \frac{w}{r+1} \rceil / w$. При $w \to \infty$ получаем искомое. □

Замечание 4. *Следующий тривиальный пример удовлетворяющий оценке леммы 2 показывает, что скорость может быть больше предельной скорости. Рассмотрим следующий код, заданный проверочной матрицей:*

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Он имеет параметры $n = 3, w = 2, r = 1$, *и его скорость* $R = 2/3$, *тогда как предельная скорость равняется* $1/2$.

4.2. Торическая топология

В этом разделе рассмотрим коды для торических топологий с $e = 1$.

Будем называть l-мерным тором решётку $n_1 \times n_2 \times \cdots \times n_l$ с зацикленными краями. Код C для двумерного тора можно получить как прямое произведение 2 кодов для циклических графов с длинами n_1, n_2. Если рассмотреть кодовые слова в виде матрицы $n_1 \times n_2$, каждая ячейка матрицы имеет две локальности – по строке и по столбцу, так как каждая строка и каждый столбец является кодовым словом кода над циклическим графом. Таким образом, скорость полученного кода равна $(\frac{r}{r+1})^2$. Далее в этом разделе будем рассматривать коды для l-мерного тора, в которых локальность задаётся множеством $N(v,r)$ из определения 5.

Лемма 3. *На* l*-мерном торе в шар радиуса* r *с центром в одной из вершин* v *входит* $S(l,r) \sim \frac{2^l r^l}{l!}$ *вершин при* $r \to \infty$.

Доказательство. Для $l = 1$ утверждение очевидно. Докажем по индукции для произвольного l.

Пусть $x_1, \ldots, x_l$ - измерения l-мерной решётки, центр шара расположим в точке $(0, \ldots, 0)$. Нужно оценить количество целочисленных решений неравенства $|x_1| + \ldots + |x_l| \leqslant r$. Неравенство $x_1 + \ldots + x_l \leqslant r$ по правилам комбинаторики имеет C^r_{r+l} неотрицательных целочисленных решений. Ровно столько вершин шара с неотрицательными координатами. Аналогичное количество получится если какие-то x_i будут неположительны. Всего есть 2^l комбинаций выбора какие координаты будут неположительны, а какие неотрицательны. Тогда если просуммировать C^r_{r+l} для каждой комбинации, то вершины, у которых некоторые координаты равны нулю, будут учтены дважды, а значит $S(l, r) < 2^l C^r_{r+l}$.

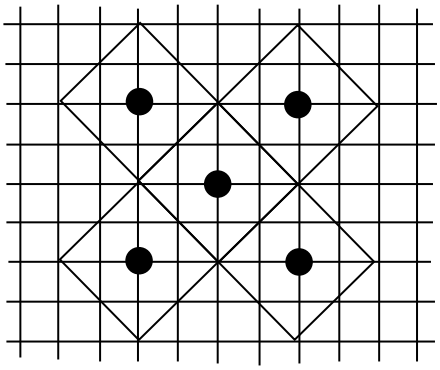
С другой стороны, по формуле включений-исключений с округлением до второго слагаемого при $r \to \infty$ получаем: $S(l, r) > 2^l C^r_{r+l} - lS(l-1, r) = 2^l C^r_{r+l} + o(r^l) = \frac{2^l r^l}{l!} + o(r^l)$, так как $S(l-1, r) = O(r^{l-1})$ по индукции. $\square$

Лемма 4. *На l-мерном торе при $n_1, \ldots, n_l, r \to \infty$:*

1) Существует множество вершин M_1 такое, что расстояние между любыми двумя вершинами этого множества превышает r и $|M_1| = l! \prod\limits_{i=1}^{l} n_i/(r^l(1 + o(1)) + o(\prod\limits_{i=1}^{l} n_i)$.

2) Существует разбиение всех вершин тора на множество M_2 подмножеств таких, что расстояние между любыми двумя вершинами одного подмножества не превышает r и $|M_2| = l! \prod\limits_{i=1}^{l} n_i/(r^l(1 + o(1)) + o(\prod\limits_{i=1}^{l} n_i)$.

Доказательство. Пусть $n_1, \ldots, n_l$ достаточно большие. Отступим от границ тора на расстояние r и будем покрывать оставшуюся область шарами в метрике Манхэттена такими, что соседние шары имеют общую грань. Ниже показан пример таких шаров радиуса 2 для $l = 2$.



Отношение области, на которую мы отступили от краёв и которую мы не замостим полностью, к $\prod_{i=1}^{l} n_i$ стремится к нулю, поэтому размер этой области $o(\prod_{i=1}^{l} n_i)$.

1) Если выбрать радиусы шаров для замощения $\lceil \frac{r+1}{2} \rceil$, то расстояние между центрами любых шаров будет превосходить r. По лемме 3 размер шара радиуса $\lceil \frac{r+1}{2} \rceil$ равен $2^l \lceil \frac{r+1}{2} \rceil^l / l! + o(r^l) = \frac{r^l}{l!} + o(r^l)$. Значит, всего шаров в покрытии будет не менее $l! \prod_{i=1}^{l} n_i / (r^l + o(r^l)) + o(\prod_{i=1}^{l} n_i)$, так как шары пересекаются. Таким образом, центры шаров являются множеством M_1 из первого утверждения леммы.

2) Если выбрать радиусы шаров для замощения $\lfloor \frac{r}{2} \rfloor$, то все точки внутри одного шара будут находиться на расстоянии не превышающем r друг от друга. По лемме 3 размер шара также будет равен $2^l \lfloor \frac{r}{2} \rfloor^l / l! + o(r^l) = \frac{r^l}{l!} + o(r^l)$. Учитывая, что размер пересечения шаров равняется $o(r^l)$ при $r \to \infty$, получаем, что шаров не более $l! \prod_{i=1}^{l} n_i / (r^l + o(r^l)) + o(\prod_{i=1}^{l} n_i)$. Примем как подмножества из второго утверждения леммы рассмотренные шары и одноэлементные множества в области на краях тора. Точки в пересечении двух шаров отнесём к любому из этих шаров. Тогда получим множество M_2 из второго утверждения леммы. □

Теорема 3. *Для предельной скорости $[n_1 \times \ldots \times n_l, k, r, w]$ LRC кода над семейством l-мерных торов при $r \to \infty$ верно:*

$\tilde{R}(r, 1, w) = 1 - l!/(r^l(1 + o(1)))$.

Доказательство. Построим граф G, вершинами которого будут являться вершины тора, а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r.

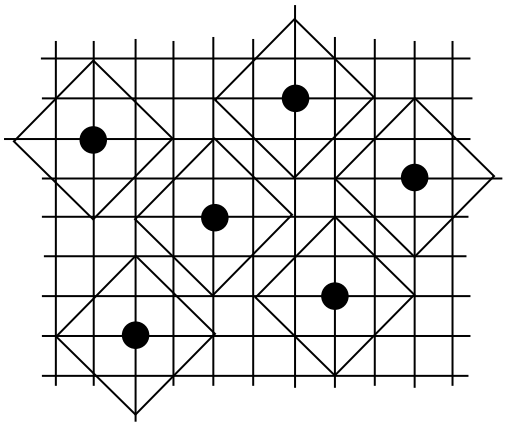
Из первого утверждения леммы 4 и леммы 2 для построенного графа G при $n_1, \ldots, n_l, r \to \infty$ получаем $\tilde{R}(r, 1, w) \leqslant 1 - l!/(r^l(1 + o(1)))$. Из второго утверждения леммы 4 и леммы 1 для графа G получаем $\tilde{R}(r, 1, w) \geqslant 1 - l!/(r^l(1 + o(1)))$. □

В утверждении ниже как и ранее требуем, чтобы размер локальности каждой вершины не превышал общего количества вершин.

Утверждение 5. *Для $l = 2$ и r кратных 2 выполнено:* $\hat{R}(r, 1, n) = \tilde{R}(r, 1, w) = 1 - \frac{2}{r^2+2r+2}$.

Доказательство. Рассмотрим следующее покрытие тора шарами радиуса $\frac{r}{2}$. Разместим шар с центром в точке (x, y) тора. Разместим четыре

шара с центрами в точках $(x+\frac{r}{2}, y-\frac{r}{2}-1)$, $(x+\frac{r}{2}+1, y+\frac{r}{2})$, $(x-\frac{r}{2}, y+\frac{r}{2}+1)$, $(x-\frac{r}{2}-1, y-\frac{r}{2})$. Далее будем продолжать так же для уже построенных шаров. Ниже показан пример такого покрытия для $l = 2$ и $r = 4$.



Построим граф G, вершинами которого будут являться вершины тора, а рёбрами будут соединены все пары вершин, расстояние между которыми не превышает r. В шар радиуса $\frac{r}{2}$ входит $\frac{r^2+2r+2}{2}$ вершин тора. Шары из рассмотренного выше замощения покрывают все вершины тора без пересечений, за исключением вершин на краях. Расстояние между центрами шаров превышает r, поэтому можно рассмотреть центры шаров покрытия как множество независимых вершин графа G для леммы 2, откуда получаем $\tilde{R}(r, 1, w) \leqslant 1 - \frac{2}{r^2+2r+2}$. С другой стороны, сами шары образуют разбиение на клики диаметра r графа G, за исключением вершин на краях. Поэтому из леммы 1 получаем $\tilde{R}(r, 1, w) \geqslant 1 - \frac{2}{r^2+2r+2}$, так как количество непокрытых вершин на краях равно $o(n_1 n_2)$.

Построим конструкцию 1 над тором. Тор является транзитивным графом, поэтому из теоремы 1 получаем, что $\hat{R}(r, 1, n) = 1 - \frac{2}{r^2+2r+2}$. □

5. Заключение

В разделе 2 были получены оценки на скорость и минимальное расстояние для произвольных графов. В разделе 3 представлена конструкция для произвольных графов с субпакетизацией. В разделе 4 были рассмотрены коды на простейших графах – цикле и торе. Были получены оценки на скорость полученной конструкции и показана оптимальность этой конструкции для некоторых типов графов. Большой интерес в литературе представляет минимальное расстояние кодов, в статье этому уделено немного внимания. В представленной конструкции субпакетизация используется для построения локального кода, в общем случае интерес

представляет также вопрос насколько субпакетизация может улучшить параметры r-локального кода.

Список литературы

[1] Ф.Дж.Мак-Вильямс, Н.Дж.А.Слоэн, *Теория кодов исправляющих ошибки*, «Связь», Москва, 1979, 744 с.

[2] S. Reed, G. Solomon, "Polynomial Codes Over Certain Finite Fields", *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 300-304

[3] I. Tamo, A. Barg, "A family of optimal locally recoverable codes", *IEEE Transactions on Information Theory*, **80**:8 (2014), 4661-4676

[4] I. Tamo, A. Barg, A. Frolov, "Bounds on the Parameters of Locally Recoverable Codes", *IEEE Transactions on Information Theory*, **62**:6 (2016), 3070-3083

[5] I. Tamo, A. Barg, S. Vladut, "Locally recoverable codes on algebraic curves", *IEEE Transactions on Information Theory*, **63**:8 (2017), 4928-4939

[6] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, S. Yekhanin, "Maximally Recoverable Codes for Grid-like Topologies", *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, 2092-2108

[7] P. Huang, E. Yaakobi, P. H. Siegel, "Multi-Erasure Locally Recoverable Codes Over Small Fields: A Tensor Product Approach", *IEEE Transactions on Information Theory*, **66**:5 (2020), 2609 - 2624

[8] A. Mazumdar, "Storage Capacity of Repairable Networks", *IEEE Transactions on Information Theory*, **61**:11 (2015), 5810 - 5821

References

[1] F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977, 762 c.

[2] S. Reed, G. Solomon, "Polynomial Codes Over Certain Finite Fields", *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 300-304

[3] I. Tamo, A. Barg, "A family of optimal locally recoverable codes", *IEEE Transactions on Information Theory*, **80**:8 (2014), 4661-4676

[4] I. Tamo, A. Barg, A. Frolov, "Bounds on the Parameters of Locally Recoverable Codes", *IEEE Transactions on Information Theory*, **62**:6 (2016), 3070-3083

[5] I. Tamo, A. Barg, S. Vladut, "Locally recoverable codes on algebraic curves", *IEEE Transactions on Information Theory*, **63**:8 (2017), 4928-4939

[6] P. Gopalan, G. Hu, S. Kopparty, S. Saraf, C. Wang, S. Yekhanin, "Maximally Recoverable Codes for Grid-like Topologies", *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017, 2092-2108

[7] P. Huang, E. Yaakobi, P. H. Siegel, "Multi-Erasure Locally Recoverable Codes Over Small Fields: A Tensor Product Approach", *IEEE Transactions on Information Theory*, **66**:5 (2020), 2609 - 2624

[8] A. Mazumdar, "Storage Capacity of Repairable Networks", *IEEE Transactions on Information Theory*, **61**:11 (2015), 5810 - 5821

Locally recoverable codes on graphs
Valinurov D.Y.

The locally recoverable codes (LRC codes) are linear codes with an important for applications property that every symbol of a codeword can be recovered from a small set of other symbols. Codeword symbols can be interpreted as servers with information. It is natural to define a topology of these servers as a graph so that every server can be recovered using neighborhood servers in this graph. The paper provides LRC code constructions for some topologies and bounds on the rate of such codes.

Keywords: erasure coding, locally recoverable codes, codes on graphs.