Вычислительная сложность определения локальности кода

Д. Ю. Валинуров[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. В статье рассматривается сведение известных NP-полных задач теории кодирования к задаче проверки свойства локальности кода, и доказывается NP-полнота данной задачи для кода над произвольным фиксированным конечным полем.

Ключевые слова: коды исправляющие ошибки, локально восстанавливаемые коды, NP-полнота.

1. Введение

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 будем рассматривать *линейные* $[n,k]$ коды[2] над конечным полем $\mathbb{F}_q$, то есть k-мерные линейные подпространства $\mathbb{F}_q^n$, где $k = \dim C$ называется *размерностью* кода C.

Порождающей матрицей линейного $[n,k]$ кода является такая матрица G размера $k \times n$, что слово $c \in C$ получается как $c = vG$, где v — некоторое слово из $\mathbb{F}_q^k$. *Проверочной матрицей* кода называется такая матрица H размера $(n-k) \times n$, что $cH^\intercal = 0$ для любого $c \in C$.

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.*

Обозначим через $[n]$ множество $\{1, 2, ..., n\}$. Для произвольного множества $A \subseteq [n]$ ограничением $H|_A$ матрицы H на A будем обозначать

[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.

[2] В дальнейшем, когда речь будет идти о линейных кодах, мы обычно будем использовать квадратные скобки вместо круглых.

подматрицу, полученную удалением из H столбцов не из A, ограничением $C|_A$ кода C на A будем обозначать код, получаемый удалением из кодовых слов всех символов с индексами 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 c_i кодового слова $c \in C$ однозначно восстанавливается по множеству R_i и является функцией компонент $c_j, j \in R_i$.

Далее также будут упомянуты линейные коды над $\mathbb{F}_q$, элементы которых являются векторами над $A = \mathbb{F}_q^w$. 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[3] как $K = kw$. Если на множестве серверов такого кода задана локальность, то соответствующий r-локальный код будем называть $[n, k, r, w]$ кодом.

Определение 3. *Кодом,* двойственным *к линейному коду $C \subseteq \mathbb{F}_q^n$ называется код $C^{\perp} = \{b \in \mathbb{F}_q^n : b \cdot c = 0, \forall c \in C\}$, где $b \cdot c = \sum_{i\in[n]} b_i c_i$ это стандартное скалярное произведение векторов $b, c \in \mathbb{F}_q^n$.*

Заметим следующую связь параметра локальности r с минимальным расстоянием d двойственного кода. Если в двойственном коде имеется кодовое слово веса d, то значит в исходном коде имеется проверочное соотношение, в котором d ненулевых коэффициентов. Поэтому видна связь задачи определения локальности с задачей определения минимального расстояния.

Определение 4. Автоморфизм *кода C — перестановка π на множестве индексов $[n]$ такая, что для любого кодого слова $c \in C$ выполнено $\pi(c) \in C$. Легко заметить, что автоморфизмы кода замкнуты относительно композиции и образуют группу, называемую* группой автоморфизмов *кода.*

[3] Для простоты мы считаем, что размерность всегда делится на w.

Определение 5. *Код называется* транзитивным*, если для любых* $i, j \in [n]$ *в группе автоморфизмов найдётся такой автоморфизм* π*, что для него выполнено* $\pi(i) = j$.

Для транзитивных кодов можно отметить следующее утверждение. Пусть d — минимальное расстояние двойственного кода, тогда из сказанного выше имеется проверочное соотношение, в котором d ненулевых коэффициентов. Вследствие транзитивности для каждого индекса i имеется проверочное соотношение с d ненулевыми коэффициентами, в которое входит индекс i, то есть локальность кода равна $d - 1$. Таким образом, для транзитивных кодов задача поиска параметра локальности равносильна задаче поиска минимального расстояния.

В статье [6] была доказана NP-полнота следующей задачи, имеющей отношение к декодированию по максимальному правдоподобию:

Задача 1. *Coset Weights.*

Дано: *Двоичная матрица* H *размера* m *x* n*, вектор* $s \in \mathbb{F}_2^m$*, целое число* $w > 0$.

Вопрос: *Существует ли вектор* $x \in \mathbb{F}_2^n$ *веса не более* w *такой, что* $Hx^\mathsf{T} = s^\mathsf{T}$*?*

В статье [7] показано, что доказательство обобщается и на случай произвольного алфавита. В статье [4] была доказана NP-полнота задачи поиска минимального расстояния сведением задачи Coset Weights. В качестве вспомогательной задачи использовалась модификация известной NP-полной задачи Subset Sum для конечного поля. Эта вспомогательная задача также будет использована далее.

Далее будет показано, что задача проверки свойства r-локальности кода в общем случае для линейного кода тоже является NP-полной. Также в статье приведён переборный алгоритм проверки свойства r-локальности.

2. Алгоритм проверки r-локальности

Рассмотрим алгоритм определения является ли код заданный проверочной матрицей H кодом с параметрами $[n, k, r, w]$. Простейший алгоритм заключается в полном переборе всех возможных локальностей для каждого i-ого сервера и проверки, что сервер i может быть однозначно восстановлен по подбираемой локальности из r серверов. Приведём ниже описание алгоритма.

Положим для начала $w = 1$. Нам потребуется вспомогательная функция CheckNodeLocality, которая по заданным матрице H, столбцу i и подмножеству столбцов $R_i \subseteq [n] \setminus i$ будет определять, является ли R_i локальностью столбца i. Обозначим через $e_i \in \mathbb{F}_q^n$ вектор с единицей в позиции i и нулями во всех остальных. Тогда псевдокод функции будет выглядеть следующим образом:

function CheckNodeLocality(H, i, R_i)
    $n \leftarrow$ число столбцов H
    $H' \leftarrow \begin{pmatrix} H \\ e_i \end{pmatrix}$
    $cols \leftarrow [n] \setminus R_i$
    return $(H'|_{\text{cols}}) = (H|_{\text{cols}})$
end function

В случае $w > 1$ вектор e_i в функции CheckNodeLocality необходимо заменить матрицей размера $w \times n$, в которой все элементы нулевые кроме единичной подматрицы $w \times w$ на месте i-ого сервера. Далее опишем основную функцию CheckFullLocality, осуществляющую перебор всех локальностей для каждого столбца:

function CheckFullLocality(H, r)
    for $i \in [n]$ **do**
        checked = false
        for $R_i \in \{A : A \subseteq [n] \setminus i, |A| = r\}$ **do**
            if CheckNodeLocality(H, i, R_i) **then**
                checked = true
                break
            end if
        end for
        if not checked **then**
            return false
        end if
    end for
    return true
end function

Правильность работы функции CheckNodeLocality основана на том, что R_i тогда и только тогда является локальностью i-ого символа, когда при занулении всех символов из R_i символ i также становится тождественно равным нулю. Это следует из того, что зависимость символа от своей локальности в случае линейного кода должна выражаться линейной комбинацией, что можно будет нетрудно видеть далее.

В CheckNodeLocality можно воспользоваться модификацией метода Гаусса, поэтому скорость работы CheckNodeLocality можно оценить сверху как $\mathcal{O}(n^3)$. Тогда CheckFullLocality содержит полный перебор и оценивается экспоненциальным временем $\mathcal{O}(n^4\binom{n-1}{r})$).

3. Доказательство NP-полноты проверки свойства r-локальности

Покажем NP-полноту проверки свойства r-локальности для линейного кода, заданного порождающей матрицей.

Задача 2. *Локальность кода C.*

***Дано**: Порождающая матрица G линейного $[n,k]$ кода C над полем $\mathbb{F}_q$. Целое положительное число $r < n$.*

***Вопрос:** Является ли C LRC кодом с параметрами [n, k, r]?*

Далее будут описаны несколько вспомогательных лемм и задач.

Лемма 1. *Дан линейный [n, k] код C. Для $i \in [n]$ существует локальность R_i тогда и только тогда, когда существует линейная комбинация, выражающая зависимость i-ого символа от символов из R_i.*

Доказательство. Пусть G - порождающая матрица кода C. Рассмотрим ограничение G на $R_i \cup \{i\}$. Обозначим $A = G|_{R_i}$, $y = G|_{\{i\}}$. Система $Ax = y$ совместна тогда и только тогда, когда $\operatorname{rank}(A|y) = \operatorname{rank}(A)$. Последнее равенство означает, что y однозначно определяется столбцами A, то есть символами из R_i, что является определением локальности. Если $Ax = y$ совместна, то её решение x является коэффициентами искомой линейной комбинации. $\square$

Задача 3. *Локальность для одного символа кода C.*

***Дано**: Порождающая матрица G линейного $[n,k]$ кода C над полем $\mathbb{F}_q$. Целое положительное число $r < n$. Целое число $i \in [n]$.*

***Вопрос:** Существует ли локальность $R_i \subseteq [n] \setminus \{i\}, |R_i| \leq r$ для индекса $i \in [n]$?*

Лемма 2. *Задача 3 полиномиально сводится к задаче 2.*

Доказательство. Без ограничения общности будем проверять r-локальность последнего символа. Из входной матрицы G получим матрицу G' дублированием в G всех столбцов кроме последнего. Очевидно, что теперь у всех символов кроме последнего имеется локальность размера один — его копия. Локальность последнего символа при этом не изменилась, поэтому применяя задачу определения r-локальности для кода заданного порождающей матрицей G' получим ответ существует ли локальность для последнего символа. $\square$

Нетрудно заметить, что задачи определения локальности 2 и 3 лежат в классе NP. Оракулу достаточно указать индексы входящие в локальность R_i для всех или для одного $i \in [n]$ соответственно, тогда модифицированным методом Гаусса можно будет проверить наличие линейной зависимости за полиномиальное время. Таким образом, если задача 3 является NP-полной, то и задача 2 является NP-полной.

Как было упомянуто, в задаче Coset Weights можно $\mathbb{F}_2$ поменять на произвольный фиксированный алфавит и требовать, чтобы s был ненулевым вектором. В качестве доказательства NP-полноты такой задачи можно без изменений использовать доказательство для задачи 1 из статьи [6]. Приведём его для полноты:

Задача 4. *Coset Weights над произвольным фикисированным конечным полем.*

Дано: *Матрица H размера $m \times n$, вектор s, $s \neq \mathbf{0}$, целое число $w > 0$.*

Вопрос: *Существует ли вектор x веса не более w такой, что $Hx^{\intercal} = s$?*

Утверждение 1. *([6]) Задача 4 для произвольного фиксированного конечного поля является NP-полной.*

Доказательство. Задача лежит в классе NP, так как при заданном x проверка свойства происходит за полиномиальное время прямым подсчётом. Будем доказывать NP-полноту сведением следующей известной задачи из статьи [3]:

Задача 5. *Трёхмерное сочетание.*

Дано: *$U \subseteq T \times T \times T$, где T - конечное множество.*

Вопрос: *Существует ли подмножество $W \subseteq U$, $|W| = |T|$ такое, что никакие два элемента W не совпадают ни в какой координате?*

Построим матрицу инцидентности H размера $3|T| \times |U|$. Каждый столбец в этой матрице содержит в точности $3T-3$ нулей и три единицы и соответствует элементу $(i,j,k) \in U$ так, что единицы в этом столбце стоят в строках с номерами i, $T+j$, $2T+k$.

Подадим задаче 4 на вход матрицу H, вектор $s = (1,1,\dots,1)$ размера $3|T|$, $w = T$. Нетрудно видеть, что вывод для такого входа совпадает с ответом на задачу Three-dimensional matching. Действительно, если существует подмножество W и x—его характеристический вектор, то $Hx^{\intercal} = s$. Обратно, если $Hx^{\intercal} = s$ для некоторого x, $|x| \leq T$, то $|x| = T$ и в сумме $\sum\limits_{\substack{1 \leq j \leq |U| \\ x_j = 1}} H_{ij} = s_i = 1$ ровно один ненулевой элемент, то есть x задаёт трёхмерное сочетание W.

$\square$

Теорема 1. *Задача 3 для произвольного фиксированного конечного поля является NP-полной.*

Доказательство. Будем сводить задачу 4 к задаче 3. На вход задачи 4 имеем набор H, s, w. В качестве порождающей матрицы G возьмём матрицу $(H|s^\intercal)$, без ограничения общности можно убрать из этой матрицы строки, линейно зависимые от других строк. Тогда по лемме 1 наличие локальности размера w у последнего символа эквивалентно существованию такого x, $|x| \leq w$, что $Hx^\intercal = s$. $\square$

4. Другое сведение к задаче проверки свойства r-локальности

На практике используются поля характеристики два, и часто матрицы в построениях используют в качестве подматрицы матрицы Вандермонда. Далее опишем другое доказательство NP-полноты задачи 3 для полей характеристики два путём сведения непосредственно задачи Coset Weights или же декодирования по максимум правдоподобия над полем $\mathbb{F}_2$. Также для задачи 3 входная матрица будет определённого вида, содержащая подматрицу Вандермонда, что можно использовать как уточнение задачи определения локальности.

В задаче Coset Weights можно рассмотреть столбцы двоичной матрицы H как вектора над $\mathbb{F}_2^m$ и представить их элементами $\mathbb{F}_{2^m}$, аналогично представляется и синдром $s \in \mathbb{F}_2^m$. Без ограничения общности можно положить, что матрица H полного ранга. Также если два столбца матрицы H, совпадают, то один из них можно убрать, при этом результат задачи не изменится. В случае нулевого вектора s решение x будет тривиальным решением, поэтому $s = 0$ можно также убрать из рассмотрения. В статье [4] похожим образом был осуществлён переход к следующей эквивалентной задаче:

Задача 6. *Finite-Field Subset Sum.*

Дано*: Целое $m \geq 2$, множество из $n \leq 2^m$ различных элементов $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{F}_{2^m}$, ненулевой элемент $\beta \in \mathbb{F}_{2^m}$ и целое положительное $r \leq m-1$.*

Вопрос: *Существует ли подмножество $\{\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_\delta}\}$ множества $\alpha_1, \alpha_2, ..., \alpha_n$ такое, что $\alpha_{i_1} + \alpha_{i_2} + ... + \alpha_{i_\delta} = \beta$ и $\delta \leq r$?*

Лемма 3. *([4]) Пусть $\alpha_1, \alpha_2, \dots, \alpha_\delta, \beta$ - различные элементы некоторого поля $\mathbb{F}_q$. Квадратная матрица M имеет следующий вид:*

$$M = \begin{pmatrix} 1 & 1 & \dots & 1 & 0 \\ \alpha_1 & \alpha_2 & \dots & \alpha_\delta & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta-2} & \alpha_2^{\delta-2} & \dots & \alpha_\delta^{\delta-2} & 0 \\ \alpha_1^{\delta-1} & \alpha_2^{\delta-1} & \dots & \alpha_\delta^{\delta-1} & 1 \\ \alpha_1^{\delta} & \alpha_2^{\delta} & \dots & \alpha_\delta^{\delta} & \beta \end{pmatrix}$$

Тогда $\det M = -(\alpha_1 + \alpha_2 + \dots + \alpha_\delta - \beta) \prod\limits_{1 \le i < j \le \delta} (\alpha_j - \alpha_i)$

Доказательство. Разложим определитель M по последнему столбцу:

$$\det M = \beta \begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_\delta \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta-2} & \alpha_2^{\delta-2} & \dots & \alpha_\delta^{\delta-2} \\ \alpha_1^{\delta-1} & \alpha_2^{\delta-1} & \dots & \alpha_\delta^{\delta-1} \end{vmatrix} - \begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_\delta \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta-2} & \alpha_2^{\delta-2} & \dots & \alpha_\delta^{\delta-2} \\ \alpha_1^{\delta} & \alpha_2^{\delta} & \dots & \alpha_\delta^{\delta} \end{vmatrix}$$

В уменьшаемом содержится определитель матрицы Вандермонда. В вычитаемом содержится выражение для альтернанта [5], для которого имеем следующее:

$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_\delta \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{j-1} & \alpha_2^{j-1} & \dots & \alpha_\delta^{j-1} \\ \alpha_1^{j+1} & \alpha_2^{j+1} & \dots & \alpha_\delta^{j+1} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta} & \alpha_2^{\delta} & \dots & \alpha_\delta^{\delta} \end{vmatrix} = S_{\delta-j}(\alpha_1, \dots, \alpha_\delta) \prod_{1 \le i < j \le \delta} (\alpha_j - \alpha_i),$$

где S_r — элементарная симметрическая функция порядка r. В частности, $S_1 = \sum\limits_{1 \le i \le \delta} \alpha_i$, откуда получаем:

$$\det M = \beta \prod_{1 \le i < j \le \delta} (\alpha_j - \alpha_i) - S_1 \prod_{1 \le i < j \le \delta} (\alpha_j - \alpha_i) =$$

$$-(\alpha_1 + \alpha_2 + \dots + \alpha_\delta - \beta) \prod_{1 \le i < j \le \delta} (\alpha_j - \alpha_i)$$

□

Утверждение 2. *Задача 6 сводится к задаче 3 над полем $\mathbb{F}_{2^m}$.*

Доказательство. Положим

$$M_\delta = \begin{pmatrix} 1 & 1 & \dots & 1 & 0 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta-2} & \alpha_2^{\delta-2} & \dots & \alpha_n^{\delta-2} & 0 \\ \alpha_1^{\delta-1} & \alpha_2^{\delta-1} & \dots & \alpha_n^{\delta-1} & 1 \\ \alpha_1^{\delta} & \alpha_2^{\delta} & \dots & \alpha_n^{\delta} & \beta \end{pmatrix}$$

Через M'_δ обозначим матрицу M_δ без последнего столбца, последний столбец обозначим m_δ.

В M'_δ выберем столбцы с некоторыми произвольными индексами $i_1, i_2, \dots, i_\delta$. Составим подматрицу $\mathfrak{M}$ размера $(\delta+1)\times(\delta+1)$ из выбранных столбцов и столбца m_δ. По лемме 1 символы с индексами $i_1, i_2, \dots, i_\delta$ образуют δ-локальность для последнего символа ($R_\delta = \{i_1, i_2, \dots, i_\delta\}$) в коде с порождающей матрицей M_δ тогда и только тогда, когда система $\mathfrak{M}x = 0$ имеет ненулевое решение $(x_1, x_2, \dots, x_{i+1})$ с $x_{i+1} \neq 0$. Если $x_{i+1} = 0$, то и все решение x нулевое. Поэтому δ-локальность последнего символа имеет место тогда и только тогда, когда $\det(\mathfrak{M}) = 0$. По лемме 3 для различных элементов $\alpha_1, \dots, \alpha_n$ условие $\det(\mathfrak{M}) = 0$ равносильно $\beta = \alpha_{i_1} + \alpha_{i_2} + \dots + \alpha_{i_\delta}$, то есть δ-локальность последнего символа эквивалентна существованию $\alpha_{i_1}, \dots, \alpha_{i_\delta}$ таких, что $\beta = \alpha_{i_1} + \alpha_{i_2} + \dots + \alpha_{i_\delta}$. Заметим, что $|R_\delta| \geq \delta$, так как если матрицу $\mathfrak{M}$ составить из меньшего количества столбцов из M'_δ, то система $\mathfrak{M}x = 0$ будет иметь только нулевое решение, а максимально возможная локальность будет $\delta+1$, когда последний столбец просто выражается через базис из $\delta+1$ столбцов матрицы Вандермонда M'_δ.

Для искомого сведения можно было бы рассмотреть все матрицы M_δ для $\delta = 1, \dots, r$. Тогда существование подмножества $\{\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_\delta}\}, \delta \leq r$ из задачи 6 равносильно тому, что хотя бы для одной матрицы M_δ получили положительный ответ на вопрос δ-локальности для последнего столбца. Но это являлось бы сводимостью по Тьюрингу, которое не обязательно означает сводимость по Карпу. Вместо этого объединим все матрицы M_δ в одну матрицу и будем получать один ответ. Сделаем это следующим образом:

$$M = \begin{pmatrix} M'_1 & 0 & \dots & 0 & m_1 \\ 0 & M'_2 & \dots & 0 & m_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & M'_r & m_r \end{pmatrix}$$

Локальность последнего символа кода с порождающей матрицей M равняется сумме локальностей последних символов в кодах с соответствующими порождающими матрицами M_i. Как было замечено, если для $\delta \in \{1, \ldots, r\}$ существует множество $\{\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_\delta}\} : \beta = \alpha_{i_1} + \alpha_{i_2} + ... + \alpha_{i_\delta}$, то последний символ кода с порождающей матрицей M_δ является δ-локальным, иначе он является $\delta + 1$-локальным. Значит, если локальность r' последнего символа кода с порождающей матрицей M равна $\sum\limits_{i \in [r]} (i+1) = (r+3)r/2$, то подмножества из задачи 6 не существует, иначе такое подмножество существует.

$\square$

5. Заключение

В разделе 2 был приведён переборный алгоритм определения локальности кода. В разделе 3 было показано, что полиномиальный алгоритм определения локальности существует только в случае если P=NP. В разделе 4 было приведено сведение оригинальной задачи Coset Weights с бинарной матрицей и задачи Finite-Field Subset Sum к задаче определения локальности.

Список литературы

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

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

[3] Michael R. Garey, David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979, 340 pp.

[4] A. Vardy, “Algorithmic Complexity in Coding Theory and the Minimum distance Problem”, *The Twenty-Ninth annual ACM symposium*, 1997, 92–109.

[5] Thomas Muir, *A Treatise On The Theory Of Determinants*, Nabu Press, 2011, 296 pp.

[6] E. Berlekamp, R. McEliece, H. van Tilborg, “On the inherent intractability of certain coding problems”, *IEEE Transactions on Information Theory*, **24**:3 (1978), 384–386.

[7] С. Барг, “Некоторые новые NP-полные задачи кодирования”, *Пробл. передачи информ.*, **30**:3 (1994), 23–28.

Computational complexity of finding code locality
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. The paper provides reductions from known decision problems of coding theory to the problem of checking such property and a proof for the NP-completeness of this problem for an arbitrary fixed finite field.

Keywords: erasure coding, locally recoverable codes, NP-complete.

References

[1] F.J. MacWilliams , N.J.A. Sloane, *The Theory of Error-Correcting Codes*, Svyaz, Moscow, 1979, 744 pp.

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

[3] Michael R. Garey, David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979, 340 pp.

[4] A. Vardy, “Algorithmic Complexity in Coding Theory and the Minimum distance Problem”, *The Twenty-Ninth annual ACM symposium*, 1997, 92–109

[5] Thomas Muir, *A Treatise On The Theory Of Determinants*, Nabu Press, 2011, 296 pp.

[6] E. Berlekamp, R. McEliece, H. van Tilborg, “On the inherent intractability of certain coding problems”, *IEEE Transactions on Information Theory*, **24**:3 (1978), 384–386

[7] S. Barg, “Some new NP-complete coding problems”, *Probl. Pered. Inform.*, **30**:3 (1994), 23–28