Теория кодирования

МФТИ, осень 2013

Александр Дайняк

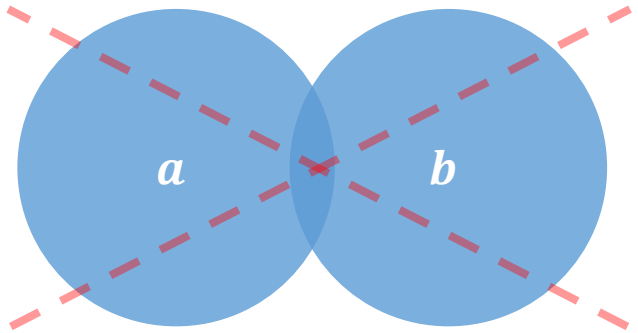
www.dainiak.com

Геометрическая интерпретация способности кода исправлять ошибки

Шар радиуса r с центром в $\boldsymbol{a}$ — это множество

$$S_r(\boldsymbol{a}) \coloneqq \{\boldsymbol{b} \mid d(\boldsymbol{a}, \boldsymbol{b}) \leq r\}$$

Если $d(C) \geq d$, то шары радиуса $\lfloor (d-1)/2 \rfloor$ с центрами в кодовых словах не пересекаются:



Задача о ближайшем кодовом слове (NCP — Nearest Codeword Problem)

Дано:

- Линейный код $C \subset \mathbb{F}_2^n$ (заданный своей порождающей матрицей)
- Слово $\boldsymbol{b} \in \mathbb{F}_2^n$

Требуется найти $\boldsymbol{a}^* \in C$, такое, что

$$d(\boldsymbol{a}^*, \boldsymbol{b}) = \min_{\boldsymbol{a} \in C} d(\boldsymbol{a}, \boldsymbol{b})$$

Задача о ближайшем кодовом слове

Теорема. Задача NCP является NP-трудной.

Для доказательства построим сведе́ния:

$$\text{3-SAT} \rightarrow \text{1-in-3-SAT} \rightarrow \text{NCP}$$

3-SAT = 3-SATISFIABILITY = 3-ВЫПОЛНИМОСТЬ

1-in-3-SAT = exactly-1-in-3 SATISFIABILITY

Задача 3-SAT

Дан набор скобок, в каждой из которых ровно три литерала (вида x или $\bar{x}$, где x — переменная)

Требуется определить, можно ли присвоить переменным значения 0 и 1, так, чтобы в каждой скобке оказался хотя бы один истинный литерал.

Задача 1-in-3-SAT

Дан набор скобок, в каждой из которых ровно три литерала (вида x или $\bar{x}$, где x — переменная)

Требуется определить, можно ли присвоить переменным значения 0 и 1, так, чтобы в каждой скобке оказался *ровно* один истинный литерал.

3-SAT $\rightarrow$ 1-in-3-SAT

Пусть дан набор скобок для задачи 3-SAT.

Заменим каждую скобку вида $(x \vee y \vee z)$ на тройку скобок $[\bar{x}, p, q][q, y, r][r, s, \bar{z}]$, так, чтобы переменные p, q, r, s нигде больше не встречались.

Например:

$$(x_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_3 \vee x_4)$$

заменится на

$$[\bar{x}_1, p_1, q_1][q_1, x_2, r_1][r_1, s_1, x_3]$$
$$[x_1, p_2, q_2][q_2, x_3, r_2][r_2, s_2, \bar{x}_4]$$

3-SAT → 1-in-3-SAT

Утверждение. В скобке вида $(x \vee y \vee z)$ есть хотя бы один истинный литерал ⇔ найдутся такие значения $p, q, r, s,$ чтобы в каждой из скобок $[\bar{x}, p, q][q, y, r][r, s, \bar{z}]$ был ровно один истинный литерал. *Доказательство — разбором случаев:*

(x, y, z)	$[\bar{x}, p, q][q, y, r][r, s, \bar{z}]$	(p, q, r, s)
(0,0,0)	$[\mathbf{1}, p, q][q, \mathbf{0}, r][r, s, \mathbf{1}]$	**?!**
(0,0,1)	$[\mathbf{1}, p, q][q, \mathbf{0}, r][r, s, \mathbf{0}]$	$(\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0})$
(0,1,0)	$[\mathbf{1}, p, q][q, \mathbf{1}, r][r, s, \mathbf{1}]$	$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$
(0,1,1)	$[\mathbf{1}, p, q][q, \mathbf{1}, r][r, s, \mathbf{0}]$	$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1})$
(1,0,1)	$[\mathbf{0}, p, q][q, \mathbf{0}, r][r, s, \mathbf{0}]$	$(\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$
(1,1,1)	$[\mathbf{0}, p, q][q, \mathbf{1}, r][r, s, \mathbf{0}]$	$(\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1})$

1-in-3-SAT $\rightarrow$ NCP

Пусть в задаче 1-in-3-SAT всего N переменных и M скобок.

Положим $n \coloneqq (M+1)(M+N)+3M,\ k \coloneqq 2N$ и укажем порождающую матрицу G линейного $[n,k]_2$-кода и слово $\boldsymbol{b} \in \mathbb{F}_2^n$.

Матрица G состоит из подматриц:

$$G \coloneqq (\underbrace{G_1|\ldots|G_1}_{(M+1)\text{ раз}}\,|\,\underbrace{G_2|\ldots|G_2}_{(M+1)\text{ раз}}\,|G_3)$$

Слово $\boldsymbol{b}$ имеет вид: $(1\ldots10\ldots0)$, количество единиц и нулей равно $(M+1)(M+N)$ и $3M$ соответственно.

1-in-3-SAT $\rightarrow$ NCP

$$n \coloneqq (M+1)(M+N)+3M, \qquad k \coloneqq 2N$$

$$G \coloneqq (\underbrace{G_1|\ldots|G_1}_{(M+1)\text{ раз}}\,|\,\underbrace{G_2|\ldots|G_2}_{(M+1)\text{ раз}}\,|G_3) \in \mathbb{F}_2^{k\times n}$$

$$G_1 \in \mathbb{F}_2^{k\times N}, \qquad G_2 \in \mathbb{F}_2^{k\times M}, \qquad G_3 \in \mathbb{F}_2^{k\times 3M}$$

- Строки матрицы G отвечают переменным и их отрицаниям.
- G_1 отвечает за выбор значений переменных.
- G_2 отвечает за то, чтобы в каждой скобке был либо один, либо три истинных литерала.
- G_3 отвечает за то, чтобы в каждой скобке было не более одного истинного литерала.

1-in-3-SAT $\rightarrow$ NCP

Матрица $G_1 \in \mathbb{F}_2^{2N \times N}$ выглядит так:

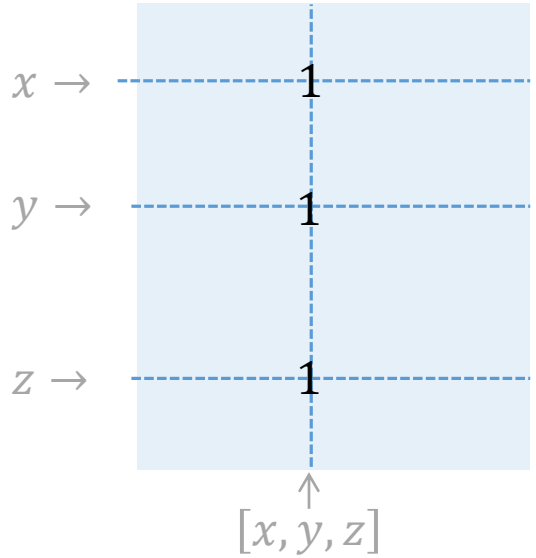
Строки G_1 соответствуют переменным задачи 1-in-3-SAT и их отрицаниям.

$$
\begin{array}{ll}
x_1 \rightarrow & 1 \quad 0 \quad \cdots \quad 0 \\
\bar{x}_1 \rightarrow & 1 \quad 0 \quad \cdots \quad 0 \\
x_2 \rightarrow & 0 \quad 1 \quad \cdots \quad 0 \\
\bar{x}_2 \rightarrow & 0 \quad 1 \quad \cdots \quad 0 \\
& \vdots \quad \vdots \quad \ddots \quad \vdots \\
x_N \rightarrow & 0 \quad 0 \quad \cdots \quad 1 \\
\bar{x}_N \rightarrow & 0 \quad 0 \quad \cdots \quad 1
\end{array}
$$

Л.к. строк G_1 равна $\mathbf{1}$ т. и т.т., когда из каждой пары строк, соответствующих одной переменной, в л.к. входит ровно одна.

1-in-3-SAT → NCP

Матрица $G_2 \in \mathbb{F}_2^{2N \times M}$ выглядит так:

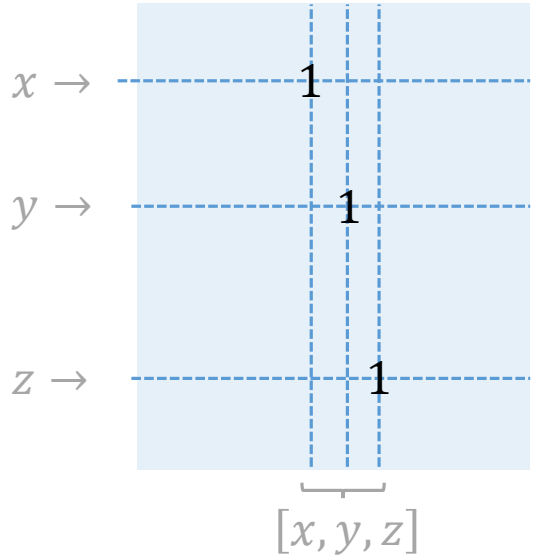


Строки G_2 соответствуют переменным задачи 1-in-3-SAT и их отрицаниям, столбцы — скобкам.

Л.к. строк G_2 равна $\mathbf{1}$ т. и т.т., когда для каждой скобки в л.к. входит ровно одна или ровно три строки, соответствующих литералам из этой скобки.

1-in-3-SAT → NCP

Матрица $G_3 \in \mathbb{F}_2^{2N \times 3M}$ выглядит так:

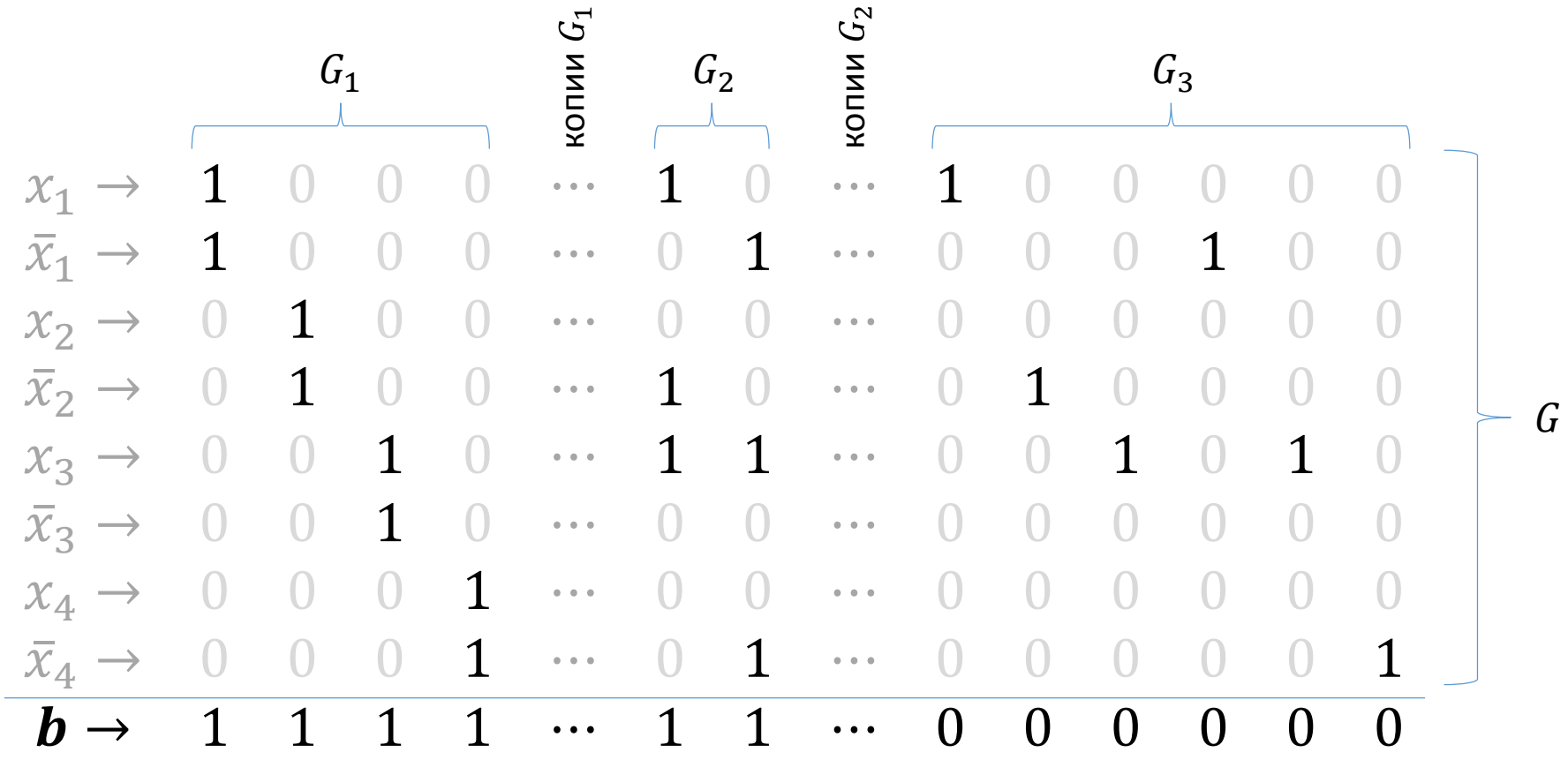


Строки G_3 соответствуют переменным и их отрицаниям. Каждой скобке отвечают три последовательных столбца.

Если для каждой скобки в л.к. строк G_3 входит ровно одна строка, соответствующая литералам из этой скобки, то вес этой л.к. равен M.

1-in-3-SAT → NCP

Пример G и $\boldsymbol{b}$ для задачи $[x_1, \bar{x}_2, x_3][\bar{x}_1, x_3, \bar{x}_4]$:



1-in-3-SAT → NCP

Пример: $[x_1, \bar{x}_2, x_3][\bar{x}_1, x_3, \bar{x}_4]$, набор значений переменных $x_1 = x_2 = 1, x_3 = x_4 = 0$.

	G_1				копии G_1	G_2		копии G_2	G_3						
$x_1 \rightarrow$	1	0	0	0	⋯	1	0	⋯	1	0	0	0	0	0	G
$\bar{x}_1 \rightarrow$	1	0	0	0	⋯	0	1	⋯	0	0	0	1	0	0	
$x_2 \rightarrow$	0	1	0	0	⋯	0	0	⋯	0	0	0	0	0	0	
$\bar{x}_2 \rightarrow$	0	1	0	0	⋯	1	0	⋯	0	1	0	0	0	0	
$x_3 \rightarrow$	0	0	1	0	⋯	1	1	⋯	0	0	1	0	1	0	
$\bar{x}_3 \rightarrow$	0	0	1	0	⋯	0	0	⋯	0	0	0	0	0	0	
$x_4 \rightarrow$	0	0	0	1	⋯	0	0	⋯	0	0	0	0	0	0	
$\bar{x}_4 \rightarrow$	0	0	0	1	⋯	0	1	⋯	0	0	0	0	0	1	
$\boldsymbol{b} \rightarrow$	1	1	1	1	⋯	1	1	⋯	0	0	0	0	0	0	

1-in-3-SAT $\rightarrow$ NCP

Утверждение. Пусть вектор $\boldsymbol{a}$ является линейной комбинацией строк построенной ранее матрицы G. Тогда $d(\boldsymbol{a}, \boldsymbol{b}) \geq M$, причём $d(\boldsymbol{a}, \boldsymbol{b}) = M$ т. и т.т., когда вектор $\boldsymbol{a}$ соответствует решению задачи 1-in-3-SAT.

(Утверждение вытекает из предыдущих рассмотрений)

Следовательно, задача 1-in-3-SAT полиномиально сводится к задаче NCP.

Задача о ближайшем кодовом слове (NCP — Nearest Codeword Problem)

Замечание. Фактически, нами доказано, что NP-трудной является задача более слабая, чем NCP: «найти минимум расстояния от заданного слова до кодовых слова заданного линейного кода».

Замечание. Можно доказать NP-трудность такой задачи: «найти максимальный вес кодовых слов заданного линейного кода»

Задача о ближайшем кодовом слове (NCP — Nearest Codeword Problem)

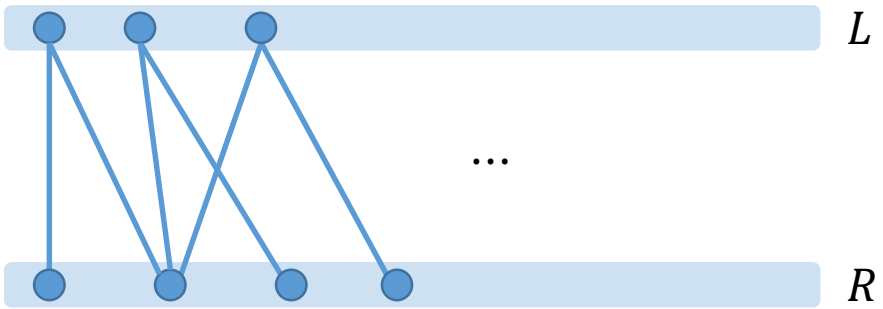
Теорема (без доказательства).
NP-трудной является следующая задача при любом фиксированном β: «для заданного линейного кода C и слова b найти такое t, что $\min\limits_{\boldsymbol{a}\in C} d(\boldsymbol{a},\boldsymbol{b}) \le t \le \beta \cdot \min\limits_{\boldsymbol{a}\in C} d(\boldsymbol{a},\boldsymbol{b})$»

(То есть решение задачи NCP трудно не только найти точно, но и приблизить с константной точностью.)

Графы-расширители (expanders)

Двудольный граф с долями L и R называется $(n, m, \Delta, \alpha, c)$-*расширителем*, если

- $|L| = n, |R| = m$
- $\deg u = \Delta$ для любого $u \in L$
- Для любого $S \subseteq L$ при $|S| \leq \alpha n$ выполнено $|N(S)| \geq c \cdot |S|$, где $N(S)$ — множество вершин в R, смежных с вершинами из S



Существование расширителей

Утверждение.
Пусть $\Delta \geq 3,\ c \leq \Delta - 2,\ \alpha < 1$ и $m \geq 4n\Delta^2\sqrt{\alpha}$.
Тогда при всех натуральных n существуют
$(n, m, \Delta, \alpha, c)$-расширители (и их много!)

Доказательство.
Построим случайный двудольный граф G и докажем,
что он с большой вероятностью будет искомым.

Существование расширителей

Зафиксируем множества L и R ($|L| = n, |R| = m$) и проведём по Δ рёбер из каждой вершины в L в выбираемые равновероятно и независимо вершины в R (в итоге некоторые из этих Δ рёбер могут попасть в одни и те же вершины R).

Рёбра G имеют естественную нумерацию, в том порядке, в котором мы определяли их концы в R (сначала Δ рёбер из 1-й вершины L, затем Δ рёбер из 2-й вершины L и т.д.).

Существование расширителей

Если G не является расширителем, то нашлось такое $S \subset L$, для которого

$$|N(S)| < (\Delta - 2) \cdot |S|$$

Оценим вероятность того, что фиксированное множество S оказалось таким «плохим» при случайном выборе концов рёбер из S в R.

А затем оценим вероятность того, что G не расширитель, по формуле

$$\Pr[G \text{ плохой}] = \Pr[\exists \text{плохое } S] \leq \sum_{S} \Pr[S \text{ плохое}]$$

Существование расширителей

Каждое ребро вида (u, v), где $u \in S, v \in N(S)$, отнесём к одному из двух типов:

- Если никакое ребро из S в $N(S)$ с меньшим номером не ведёт в v, то ребро (u, v) назовём «*первопроходцем*»
- В противном случае, назовём (u, v) «*дублем*»

Очевидно, всего будет $N(S)$ «первопроходцев» и $(\Delta \cdot |S| - |N(S)|)$ «дублей».

Существование расширителей

Всего $N(S)$ «первопроходцев» и $(\Delta \cdot |S| - |N(S)|)$ «дублей».

Мы предполагаем, что $|N(S)| < (\Delta - 2) \cdot |S|$, а значит «дублей» будет не менее $2 \cdot |S|$.

Обозначим $s \coloneqq |S|$. Вероятность того, что среди рёбер из S в $N(S)$ есть $2s$ дублей

$$\leq \binom{\Delta \cdot s}{2s} \left(\frac{\Delta \cdot s}{m}\right)^{2s}$$

Число способов выбрать рёбра-«дубли»

Верхняя оценка вероятности попадания конца «дубля» в одну из вершин в $|N(S)|$

Существование расширителей

Вероятность того, что заданное множество $S \subset L$ «плохо расширяется», не превосходит

$$\binom{\Delta \cdot |S|}{2|S|}\left(\frac{\Delta \cdot |S|}{m}\right)^{2|S|}$$

Значит,

$$\Pr[G \text{ не расширитель}] \leq \sum_{\substack{S \subset L \\ 1 \leq |S| \leq \alpha n}} \Pr[S \text{ «плохое»}] \leq \sum_{1 \leq s \leq \alpha n} \binom{n}{s}\binom{\Delta \cdot s}{2s}\left(\frac{\Delta \cdot s}{m}\right)^{2s}$$

Существование расширителей

С учётом оценки $\binom{a}{b} < \left(\frac{ea}{b}\right)^b$ получаем

$$\Pr[G \text{ не расширитель}] \leq \sum_{1 \leq s \leq \alpha n} \binom{n}{s} \binom{\Delta s}{2s} \left(\frac{\Delta s}{m}\right)^{2s} \leq$$

$$\leq \sum_{1 \leq s \leq \alpha n} \left(\frac{en}{s}\right)^s \left(\frac{e\Delta s}{2s}\right)^{2s} \left(\frac{\Delta s}{m}\right)^{2s} = \sum_{1 \leq s \leq \alpha n} \left(\frac{e^3 \Delta^4 sn}{4m^2}\right)^s \leq \sum_{1 \leq s \leq \alpha n} \left(\frac{e^3 \Delta^4 \alpha n^2}{4m^2}\right)^s$$

Существование расширителей

$$\Pr[G \text{ не расширитель}] \leq \sum_{1 \leq s \leq \alpha n} \left(\frac{e^3 \Delta^4 \alpha n^2}{4m^2} \right)^s$$

При $m \geq 4n\Delta^2\sqrt{\alpha}$ выполняется неравенство

$$\frac{e^3 \Delta^4 \alpha n^2}{4m^2} < \frac{1}{3}$$

Отсюда

$$\Pr[G \text{ не расширитель}] \leq \sum_{1 \leq s \leq \alpha n} \left(\frac{1}{3} \right)^s < \frac{1}{2}$$

Существование расширителей

Итак, случайный двудольный мультиграф будет расширителем с вероятностью не менее $\frac{1}{2}$.

Чтобы от мультиграфа перейти к обычному графу, достаточно перенаправить концы рёбер-дублей в произвольные вершины R.

Свойства расширительности от этого могут только улучшиться.

Коды на основе двудольных графов

Код на основе двудольного графа — это линейный двоичный код, строящийся так:

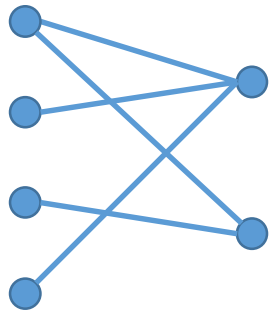
- Вершинам из L соответствуют переменные $x_1, \dots, x_n$
- Вершинам из R соответствуют уравнения: если в вершину $v \in R$ входят рёбра из вершин $u_{i_1}, \dots, u_{i_l}$, то уравнение будет

$$x_{i_1} + \cdots + x_{i_l} = 0$$

Искомый код состоит из всех слов $(x_1 \dots x_n)$, удовлетворяющих системе этих уравнений.

Коды на основе двудольных графов

Например, для графа



соответствующий код будет выглядеть так:

$$\{(x_1x_2x_3x_4) \mid x_1 + x_2 + x_4 = 0, \quad x_1 + x_3 = 0\}$$

Коды на основе двудольных графов

Утверждение.
Код, построенный по двудольному графу, в котором $|L| = n$ и $|R| = m$, является двоичным линейным $[n, k, d]$-кодом, где $k \geq n - m$.

Доказательство:

Код является множеством решений системы из m уравнений с n неизвестными.

Значит, он образует линейное пространство размерности не менее чем $n - m$.

Коды на основе расширителей: кодовое расстояние

Теорема. (M. Sipser, D.A. Spielman)
Если $c > \frac{\Delta}{2}$, и C — код, построенный на основе $(n, m, \Delta, \alpha, c)$-расширителя, то $d(C) > \alpha n$.

Доказательство: от противного.

Допустим, что $d(C) \leq \alpha n$. Тогда найдётся слово $\boldsymbol{a} \in C$, такое, что $\boldsymbol{a} \neq \mathbf{0}$ и $\|\boldsymbol{a}\| \leq \alpha n$.
Пусть $I \coloneqq \{u_1, \dots, u_{\|\boldsymbol{a}\|}\}$ — вершины из L, соответствующие единичным координатам $\boldsymbol{a}$.

Коды на основе расширителей: кодовое расстояние

Т.к. наш граф — расширитель, и $\|\boldsymbol{a}\| \leq \alpha n$, то

$$|N(I)| \geq c \cdot |I| > \frac{\Delta}{2} \cdot |I|$$

Всего из I в $N(I)$ ведёт ровно $\Delta \cdot |I|$ рёбер.

Поэтому *среднее* число рёбер, входящее в вершины $N(I)$ из I, равно

$$\frac{\Delta \cdot |I|}{|N(I)|} < \frac{\Delta \cdot |I|}{\frac{\Delta}{2} \cdot |I|} = 2$$

Значит, в $N(I)$ найдётся вершина, в которую входит ровно одно ребро из I.

Коды на основе расширителей: кодовое расстояние

В $N(I)$ найдётся вершина, в которую входит ровно одно ребро из I.

Значит, среди задающих код уравнений есть такое уравнение $x_{i_1} + \cdots + x_{i_l} = 0$, в котором ровно одна из переменных на слове $\boldsymbol{a}$ обращена в единицу.

Но этого не может быть в предположении, что $\boldsymbol{a}$ является решением этого уравнения. Противоречие.

Коды на основе расширителей: исправление ошибок

Теорема. (M. Sipser, D.A. Spielman)
Пусть $c > \frac{3\Delta}{4}$, и C — код, построенный на основе $(n, m, \Delta, \alpha, c)$-расширителя.

Пусть слово $\boldsymbol{a}'$ получено из некоторого кодового слова $\boldsymbol{a}$ искажением не более чем $\frac{\alpha n}{4}$ битов.

Тогда восстановить $\boldsymbol{a}$, зная $\boldsymbol{a}'$, можно с помощью следующего алгоритма…

Коды на основе расширителей: исправление ошибок

Алгоритм Сипсера—Шпильмана:

1. Если $\boldsymbol{a}' \in C$, то выводим $\boldsymbol{a}'$ и завершаем работу.
2. Если $\boldsymbol{a}' \notin C$, то для $\boldsymbol{a}'$ некоторые из уравнений (отвечающих вершинам в R) нарушены. Считаем поочерёдно для каждого бита $\boldsymbol{a}'$ число нарушенных уравнений, в которых он участвует. Если их $> \Delta/2$, инвертируем этот бит и идём на шаг 1.

Коды на основе расширителей: исправление ошибок

Лемма «о результативном бите».

Пусть $c > \frac{3\Delta}{4}$, и C — код, построенный на основе $(n, m, \Delta, \alpha, c)$-расширителя.

Пусть $\boldsymbol{a}' \notin C$, но при этом $d(\boldsymbol{a}, \boldsymbol{a}') \leq \alpha n$ для некоторого $\boldsymbol{a} \in C$.

Тогда в $\boldsymbol{a}'$ найдётся бит, обращение которого на противоположный строго уменьшает число невыполненных для $\boldsymbol{a}'$ уравнений. (Имеются в виду уравнения, построенные по графу-расширителю)

Коды на основе расширителей: исправление ошибок

Пусть $\boldsymbol{a}'$ — не кодовое слово, находящееся от ближайшего кодового на расстоянии $\leq \alpha n$.

Пусть $I \subset L$ — множество вершин, соответствующих координатам, в которых $\boldsymbol{a}'$ отличается от ближайшего кодового слова.

Обозначим через $N_{\text{pass}}(I)$ вершины из $N(I)$, соответствующие уравнениям, выполненным на слове $\boldsymbol{a}'$.

Аналогично $N_{\text{fail}}(I)$ — вершины из $N(I)$, отвечающие нарушенным уравнениям.

Коды на основе расширителей: исправление ошибок

- $N(I) = N_{\text{pass}}(I) \sqcup N_{\text{fail}}(I)$

Так как наш граф расширитель, и $|I| \leq \alpha n$, то

$$\left|N_{\text{pass}}(I)\right| + |N_{\text{fail}}(I)| = |N(I)| \geq c|I| > \frac{3\Delta}{4} \cdot |I|$$

Из каждой вершины $N(I)$ в I ведёт хотя бы одно ребро.

При этом из каждой вершины $N_{\text{pass}}(I)$ в I ведёт хотя бы два ребра (чтобы «обмануть» уравнение, нужно инвертировать в нём чётное количество переменных)!

Коды на основе расширителей: исправление ошибок

- $\left|N_{\text{pass}}(I)\right| + |N_{\text{fail}}(I)| > \frac{3\Delta}{4} \cdot |I|$
- $\#\left\{\text{рёбер из } N_{\text{pass}}(I) \text{ в } I\right\} \geq 2 \cdot \left|N_{\text{pass}}(I)\right|$
- $\#\{\text{рёбер из } N_{\text{fail}}(I) \text{ в } I\} \geq |N_{\text{fail}}(I)|$
- Число рёбер между I и $N(I)$ равно $\Delta \cdot |I|$.

Из всего этого выводим

$$\Delta \cdot |I| \geq |N_{\text{fail}}(I)| + 2 \cdot \left|N_{\text{pass}}(I)\right| > |N_{\text{fail}}(I)| + 2 \cdot \left(\frac{3\Delta}{4} \cdot |I| - |N_{\text{fail}}(I)|\right)$$

Отсюда $|N_{\text{fail}}(I)| > \frac{\Delta}{2} \cdot |I|$

Коды на основе расширителей: исправление ошибок

Итак, общее число нарушенных уравнений, в которых участвуют вершины-переменные из I, *строго больше* чем $\frac{\Delta}{2} \cdot |I|$.

Значит, в I найдётся вершина, для которой нарушены *больше* половины тех уравнений, в которых она участвует.

То есть, даже не зная I, можно утверждать следующее: *среди координат* $\boldsymbol{a}'$ *есть хотя бы одна такая, обратив значение которой мы уменьшим число нарушенных уравнений.*

Коды на основе расширителей: исправление ошибок

Лемма о результативном бите говорит, что если к очередному шагу алгоритма мы пришли с некодовым словом $\boldsymbol{a}'$, находящимся от ближайшего кодового на расстоянии $\leq \alpha n$, то очередной бит для изменения мы найдём.

Осталось доказать, что, начав со слова $\boldsymbol{a}'$ на расстоянии $\leq \frac{\alpha n}{4}$ от ближайшего кодового слова $\boldsymbol{a}$, мы не «притянемся» случайно к какому-то другому кодовому слову $\boldsymbol{b} \neq \boldsymbol{a}$.

Коды на основе расширителей: исправление ошибок

До начала работы алгоритма $|I| \le \frac{\alpha n}{4}$, и значит

$$\#\text{нарушенных уравнений} \le \Delta \cdot |I| \le \frac{\alpha n \Delta}{4}$$

В ходе работы алгоритма число нарушенных уравнений уменьшается.

Пусть на очередном шаге получено слово $\boldsymbol{a}''$, и пусть I'' — биты, в которых $\boldsymbol{a}''$ отличается от ближайшего кодового слова. Имеем

$$\frac{\alpha n \Delta}{4} \ge \#\text{наруш. ур.} = N_{\text{fail}}(I'') > \frac{\Delta}{2} \cdot |I''|$$

отсюда $|I''| < \frac{\alpha n}{2}$.

Коды на основе расширителей: исправление ошибок

Итак, на каждом шаге алгоритма получаем слово, отличающееся от ближайшего кодового менее чем в $\frac{\alpha n}{2}$ битах.

Т.к. на каждом шаге в слове меняется только один бит, и $d(C) > \alpha n,$ то кодовое слово, к которому мы стремимся, всё время одно и то же.

(Т.к., если $d(\boldsymbol{a}, \boldsymbol{b}) > t$, то, находясь в шаре $S_{t/2}(\boldsymbol{a})$ и смещаясь на один бит, мы не вывалимся в шар $S_{t/2}(\boldsymbol{b})$)