ISSN 2225-9309 (Print)
2025. № 2 (33). С. 109–113
Молодежный Вестник УГАТУ
http://journal.ugatu.su

УДК 004.65
doi 10.54708/22259309_2025_233109

Словари кодов с фиксированным порогом различия

Ф. А. Трошков[1]

[1] fedortroshkov@yandex.ru

[1] ФГБОУ ВО «Уфимский университет науки и технологий» (УГАТУ)

Аннотация. В статье предложен алгоритм построения словарей кодовых слов по единому шаблону, удовлетворяющих требованию «порога различия». Под порогом различия подразумевается минимальное количество различных внутри любой пары слов из словаря позиций в рамках шаблона, под которым, в свою очередь, подразумевается произвольно заданная конечная упорядоченная последовательность конечных множеств-«алфавитов», представляющих собой возможные буквы на каждой позиции в слове. Рассматривается связь описанного метода кодирования и построения систем ортогональных латинских квадратов, а также применение «обобщённой шахматной ладьи» для визуализации соотношений между кодами в словаре.

Ключевые слова: код; кодирование; равномерное кодирование; исправление ошибок; комбинаторика; латинский квадрат; ортогональные латинские квадраты.

ВЕДЕНИЕ

Назовем шаблоном кодирования конечный упорядоченный набор конечных множеств, элементы которых будем считать символами или буквами. Словом, составленным по шаблону, назовём упорядоченный набор символов, каждый из которых выбран из множества, соответствующего ему по номеру. Введём метрику различия для пары слов, равную количеству позиций шаблона, для которых выбранные символы у проверяемой пары слов отличаются (расстояние Хэмминга). Порогом различия для словаря будет минимальное значение метрики по исследуемому словарю. В данной статье рассматривается способ составить словарь по заданным шаблону и порогу.

Ключевые идеи. Представив пространство слов в виде многомерной таблицы (координатные оси соответствуют множествам из шаблона), получим клетки, в каждой из которых находится одно слово, а любые два слова, стоящие в разных клетках, различаются минимум на одну букву. Поднимая порог до двух, обнаружим, что выбор произвольной ячейки в таблице запрещает выбирать ячейки, лежащие на проходящих сквозь неё осях – однопараметрических дискретных пространствах. Это соответствует поведению шахматной ладьи – она перемещается и атакует строго в проходящих сквозь неё столбце и строке, то есть фигура соперника, делящая с ней одну координату, попадает под удар. Назовём это первым «уровнем» и сразу обобщим: ладья уровня n атакует все фигуры, отстоящие от неё не более, чем по n осям, а различающиеся по n+1 осям она оставит в покое. Таким образом, ладья уровня 2 будет «съедать» проходящие через неё плоскости, параллельные осям координат, ладья уровня 3 будет «съедать» трёхмерные подпространства и так далее.

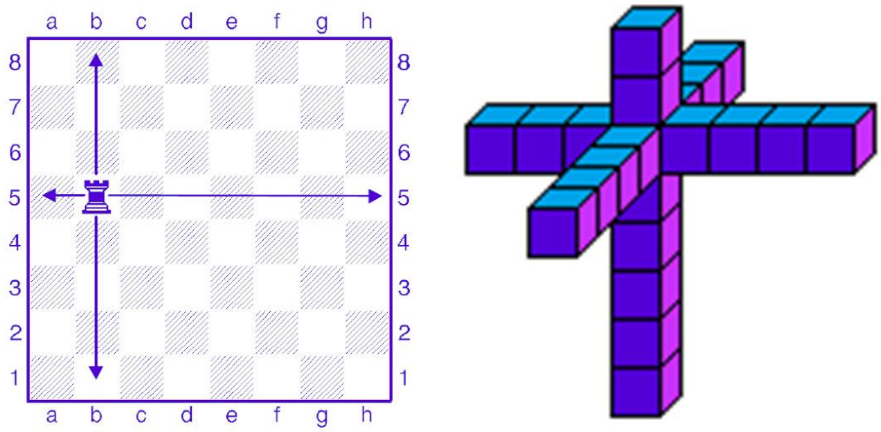

Рис. 1. Ход ладьи на шахматной доске и «съеденные» оси в трёхмерном пространстве

Для ладей любого уровня справедливо наблюдение: так как у ладьи нет других ладей на проходящей сквозь неё оси, параллельной координатной, при проецировании пространства вдоль этой оси изображения ладей не наложатся друг на друга. Далее можно записать в клетке, куда попадает спроецированная ладья, её координату на оси проекции. После нескольких повторных проекций получится таблица с меньшей размерностью и упорядоченным набором чисел в каждой клетке – количество чисел в каждой клетке равно уровню используемых ладей. При этом правила расстановки сохраняются: две разные ладьи должны отличаться друг от друга хотя бы по n+1 осям, только теперь к видимым осям проекции добавляются скрытые оси, вдоль которых исходное пространство было спроецировано, и координаты на которых теперь записаны в клетках.

ЛАТИНСКИЕ ПРОЕКЦИИ

Возьмём шаблон, состоящий из трёх позиций с равным количеством букв для каждой позиции. Потребуем различия между словами в две буквы, то есть сожмём одну ось при проецировании пространства. Получится двумерная таблица, содержащая по числу в клетке. В силу равенства осей таблицы числа в клетках имеют тот же диапазон, что и координаты по двум видимым осям.

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

Рис. 2. Двумерная проекция трёхмерного словаря с порогом 1

Распределение этих чисел таково, что в двух клетках с общими строкой или столбцом числа должны различаться. Расстановка максимального количества ладей в таком пространстве соответствует структуре, называемой латинским квадратом. Латинский квадрат – это таблица n × n, в каждой клетке которой находится один из n вариантов объекта (букв или чисел). Объекты расположены так, что в каждом столбце и каждой строке все варианты встречаются по одному.

Это построение можно продолжить для составления словарей с различием в две буквы в пространствах любой размерности. Спроецируем таблицу вдоль самой длинной оси. Выбрав вторую по длине ось, заполним одно из параллельных ей одномерное подпространство по порядку всеми числами, какие поместятся (так как первая ось по длине может быть равна или длиннее). Затем сделаем шаг вдоль другой оси и скопируем первое подпространство в новое, добавив к каждой позиции 1 (зацикливая в кольце остатков по модулю длины проективной оси). Сделав ещё шаг по той же оси, перенесём туда предыдущее подпространство и также прибавим 1. Продолжать это копирование можно до конца оси, по которой идём, так как она не длиннее оси проекции, иначе пришлось бы остановиться, прежде чем зацикливание чисел привело к появлению копии исходного подпространства. Так, заполнив уже двумерное пространство, начнём копировать его вдоль ещё одной оси, также прибавляя 1 к каждой клетке копии. Так можно поставить максимальное количество ладей первого уровня в пространстве. Подходом, равносильным этому, будет заполнение клеток проекции значениями функции: остатками от деления суммы координат на длину оси проекции (если спроецировать вдоль длиннейшей оси).

$$X \widetilde{= (} x_1, x_2, \dots, x_3) \in \{1..A_1\} \times \{1..A_2\} \times \dots \times \{1..A_n\}$$

$$f(\tilde{X}) = \left(\sum x_i\right) mod A_{max}$$

Так как значение функции является координатой по единственной скрытой оси для ладьи на проекции, главное требование к нему – отличаться в случае отличия двух аргументов только в одной позиции.

$$f(\tilde{X}, x_k = p) \neq f(\tilde{X}, x_k = q)$$

Так как в качестве делителя для нахождения остатков выбрано наибольшее измерение таблицы (ось проекции – самая длинная), остатки не могут зациклиться:

$$f(\tilde{X}, x_k = 1) \neq f(\tilde{X}, x_k = 2) \neq f(\tilde{X}, x_k = A_k)$$

$$A_k < A_{max} \rightarrow x_k < A_{max},$$

то есть остатки не повторяются.

Функция заполнения имеет ещё одно замечательное свойство:

$$\mathrm{a} \in \{0..A_{max} - 1\}: \quad f_a(\tilde{X}) = \left(a + \sum \mathrm{x_i}\right) \mathrm{mod} A_{max}.$$

Прибавляя один из возможных остатков, можно получить столько вариантов словаря, сколько символов есть в наибольшем множестве из шаблона. Внутри каждого словаря будет наблюдаться различие в две буквы между словами.

ГРЕКО-ЛАТИНСКИЕ ПРОЕКЦИИ

Сочетание требования к различию в три символа (ладьи второго уровня) и шаблона из четырёх букв (четырёхмерное пространство) в проекции соответствует структуре, называемой греко-латинским или двойным латинским квадратом: в каждой клетке находятся два числа, нет совпадающих клеток, для первых и вторых чисел по отдельности выполняются правила обычного латинского квадрата (также называется парой ортогональных латинских квадратов). Заполнение таких структур возможно, кроме случаев 2 × 2 и 6 × 6 (наибольшее число ладей – 34) [1].

Дальнейшие построения требуют нахождения уже не пар, а троек и более объёмных систем попарно ортогональных латинских квадратов.

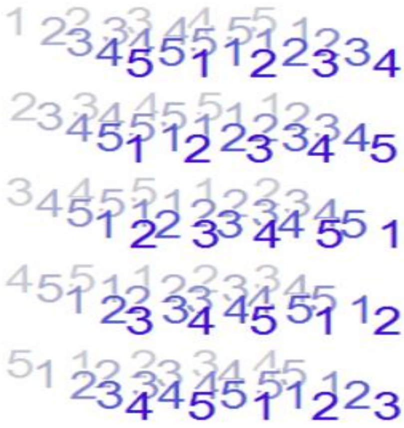

Рис. 3. Трёхмерная проекция четырёхмерного словаря с порогом 1

ВЫРЕЗАНИЕ ИЗ ГИПЕРКУБА

Представляя всё пространство слов как таблицу и проецируя её по стольким осям, сколько «съедают» используемые ладьи, несложно убедиться, что потолком количества ладей будет количество клеток в самой маленькой проекции.

Таким образом, заполняемость пространства зависит от заполняемости этой маленькой проекции, которую можно представить как фрагмент гиперкубической таблицы, содержащей в клетках упорядоченные наборы чисел в диапазонах, равных измерению. Измерением гиперкуба будет наибольшее измерение исходной таблицы. Теперь пространство можно заполнить максимально, если соответствующий гиперкуб можно заполнить полностью или так, что противоречивые клетки окажутся отрезаны.

Максимальное заполнение словарей с различием слов в трёх символах упрощается в случае, если проекцию можно вырезать из гиперкуба с измерением – простым числом, то есть два больших измерения равны одному простому числу.

Гиперкуб размерности 4 заполняется в проекции построчно. Первая строка – произвольно, а каждая следующая – циклическим сдвигом первых чисел на a, вторых чисел – на b, где a и b – взаимно простые.

Для создания куба размерности 5 можно накладывать копии заполненной проекции куба размерности 4 друг на друга, циклически сдвигая по двум измерениям.

Далее подобными наложениями со сдвигом по двум осям создаются гиперкубы последующих размерностей.

ЗАКЛЮЧЕНИЕ

В статье рассмотрена проблема составления словарей из кодов одинаковой длины, удовлетворяющих порогу числа различных символов в каждой паре, и предложены алгоритмы для порогов в 1, 2 и 3 различия.

СПИСОК ЛИТЕРАТУРЫ

1. **Ahuja R.** Error-correcting codes and latin squares [Электронный ресурс]. https://ijarse.com/images/fullpdf/1517035466_J1016ijarse.pdf (дата обращения 01.11.2024)

2. **Dénes J., Keedwell A. D.** Latin squares and their applications. New York-London: Academic Press. P. 547.

3. **Gardner Martin.** Martin Gardner's New Mathematical Diversions from Scientific American. Fireside.

ОБ АВТОРЕ

ТРОШКОВ Фёдор Антонович, студент 2 курса специальности Прикладная математика и информатика (УУНиТ)

METADATA

Title: Code dictionaries with fixed minimal Hamming distance.

Author: F. A. Troshkov[1]

Affiliation:

[1] Ufa University of Science and Technology (UUST), Russia.

Email: [1] fedortroshkov@yandex.ru.

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa University of Science and Technology), no. 2 (33), pp. 109-113, 2025. ISSN 2225-9309 (Print).

Abstract: Construction of block codes with respect to a given threshold Hamming distance is discussed. Algorithm for constructing such codes for 1, 2 or 3 units of minimal distance are described.

Key words: coding, block code, combinatorics, Latin square, orthogonal Latin squares.

About authors:

TROSHKOV, Fyodor Antonovich, student, Applied mathematics and informatics (UUST)