БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ПАСКАЛ НИКОЛАЕВ ПИПЕРКОВ

ДИСКРЕТНИ ТРАНСФОРМАЦИИ И ПРИЛОЖЕНИЕТО ИМ В ТЕОРИЯ НА КОДИРАНЕТО И КОМБИНАТОРИКАТА

Д И С Е Р Т А Ц И Я

за присъждане на образователна и научна степен „доктор“

Професионално направление: 4.5. Математика

Научна специалност:
01.01.02. Алгебра и теория на числата

Научен ръководител:
проф. дмн Илия Буюклиев

Велико Търново
2022 г.

С Ъ Д Ъ Р Ж А Н И Е

Увод

В разработката са описани дискретни трансформации и някои техни приложения за намиране на параметри на линейни кодове. Основна роля за ефективността на изчислителните алгоритми играят бързите трансформации, разработени през 60-те години на XX век. Макар широко разпространени в шумозащитното кодиране и обработката на сигнали, бързите дискретни трансформации имат все още неизползван потенциал за прилагане в различните области на науката и техниката.

Същността на някои дискретни трансформации (преобразувания) е умножението на вектор с матрица. Спецификата на конкретна трансформация се изразява в типа на използваната матрица. За целите на бързите алгоритми основната матрица се представя като произведение на разредени матрици, чиито редове се състоят от нули с изключение на малък брой елементи, например със стойност 1 или -1. Това води до алгоритми с по-малка сложност, отколкото при обичайното умножение на матрица с вектор [34, 42, 65]. Обзор на бързи трасформации и техни приложения е направен в [17, 39, 53, 67].

В исторически план, функциите на Уолш възникват като дискретен аналог на ортонормираната система от тригонометрични функции, а трансформацията на Уолш-Адамар – като аналог на преобразуването на Фурие [71]. Удобството на функциите на Уолш е, че са стъпаловидни, което от една страна дава възможности за директно преминаване към дискретна трансформация, а от друга страна запазва познатите от преобразуването на Фурие свойства.

Функциите на Уолш и свойствата на техните редове от тип Фурие са изследвани от Пели [61], Файн [41], Моргенталер [59], Хармут [46] и др. Трансформацията на Уолш-Адамар е разглеждана независимо под различни наименования от Колеман [33], Лехнер [54], Онсорг [6] и др. Връзката с функциите на Уолш установява Хендерсон [48]. Трансформацията намира приложения в спектралния анализ [8], за обработка на аудио и видео сигнали, изображения [7, 63], реч [66]. Интересът към функциите на Уолш и съответната трансформация провокира симпозиуми през 1970–1974 г. [1, 2, 3, 4, 5]. Трансформацията на Уолш-Адамар се прилага при изследване на комбинаторни конфигурации като булеви и вектор булеви функции [26, 27], двоични линейни кодове [51] и

други.

Функциите на Виленкин-Крестенсон [28, 70] и съответната трансформация са обобщение на функциите и трансформацията на Уолш в комплексни числа, като за основа вместо -1 се взема q-ти примитивен комплексен корен на единицата. Трансформацията е приложима за комбинаторни конфигурации над крайни прости полета [51, 52].

За линейни кодове над съставни крайни полета е удобно да се ползва трансформацията на следите [9, 52]. При нея за основа се взема примитивен комплексен корен на единицата от степен характеристиката на полето. При изчисленията, вместо скаларно произведение, се ползва следата на скаларното произведение.

Задачата, поставена за решаване чрез дисертационния труд, е намиране на ефективни алгоритми за изчисляване на тегловното разпределение и радиуса на покритие на линеен код над крайно поле чрез използване на характеристичен вектор.

Линейните кодове се дефинират като линейни подпространства на n-мерното линейно пространство над крайно поле. Те се конструират и ползват в термините на пораждаща матрица, чиито редове са базис на подпространството. Намирането на параметрите на кода (тегловно разпределение, минимално разстояние, радиус на покритие) по дадена пораждаща (или проверочна) матрица са основни и важни задачи в много аспекти от теория на кодирането. Установяването (поправянето) на грешки при пренос на информация е една от целите в теорията на кодирането, а определящ фактор за това са минималното тегло и радиусът на покритие на избрания линеен код. Радиусът на покритие играе роля в определянето дали кодът е съвършен, квазисъвършен, допуска ли възможност за разширяване и т. н. Теория на кодирането е систематически изградена, например в [13, 16, 49, 57, 62].

Известно е, че задачата за намиране на тегловно разпределение е от клас NP-пълни задачи [14]. Независимо от това, разработени са множество алгоритми за пресмятане на тегловно разпределение. Някои от тях са вложени в софтуерни системи, свързани с теория на кодирането, като MAGMA [20], GUAVA [10], Q-Extension [21] и други. Основната идея на общите алгоритми е да се получат всички линейни комбинации на базисните вектори и да се пресметнат техните тегла. Ефективните алгоритми генерират всички кодови думи в редица, в която всяка кодова дума се получава от предишната с прибавяне на една кодова дума. Те обикновено ползват q-ични кодове на Грей [19, 44] или допълнителни матрици [23]. Сложността на тези алгоритми е $O(nq^k)$.

Задачата за намиране на радиус на покритие на линеен код е пълна в класа Π_2^p в полиномиалната йерархия [58]. Изследователите търсят долна и

горна граница за радиуса на покритие за определени класове линейни кодове [29, 30, 31, 32, 43]. В тези проучвания активно се включва групата по теория на кодирането в България [11, 12, 25, 35, 36, 37, 38].

Повод за проучването са резултатите и идеите на Марк Карповски [51, 52] за прилагане на бързи дискретни трансформации за намиране на тегловното разпределение и радиуса на покритие на линеен код. За двоични линейни кодове Карповски прилага трансформацията на Уолш-Адамар. Като обобщение при недвоични линейни кодове предлага използването на трансформацията на Виленкин-Крестенсон (за прости полета) и трансформацията на следите (за съставни полета).

Съществен принос на настоящата разработка е, че изчисленията се правят върху максимално множество от непропорционални вектори. Това от една страна е достатъчно за определяне на тегловната функция и други параметри на линейния код, а от друга страна намалява сложността на алгоритмите и обема на ползваната памет. За целта пораждащата (проверочната) матрица, която задава линейния код, се представя чрез характеристичен вектор на нейните стълбове, който отчита броя на стълбовете, принадлежащи към съответните точки на проективна геометрия. Конструкцията е полезна, когато броят на редовете на матрицата е значително по-малък от броя на нейните стълбове.

Глава 1 е посветена на базови понятия. В раздел 1 са въведени някои понятия и твърдения за крайни полета – следа, самодуален базис и др. В раздел 2 са описани основни понятия и твърдения за линейни кодове. В раздел 3 е въведена трансформацията на Уолш-Адамар. В раздел 4 са дадени методи за представяне на Кронекерова степен като произведение на разредени матрици. Тази техника стои в основата на бързите трансформации и съответните бътерфлай алгоритми. В раздел 5 е описана трансформацията на Виленкин-Крестенсон. В раздел 6 е описана трансформацията на следите.

В **Глава 2** е описан разработен алгоритъм за пресмятане на тегловно разпределение на линеен код над крайно просто поле. В раздел 1 е описан специален тип на пораждащата матрица на симплекс кода и е дефинирано понятието характеристичен вектор по отношение на симплекс кода. В раздел 2 е дефинирано понятието характеристично разпределение и са изведени неговите свойства. Показана е връзката с тегловното разпределение. В раздел 3 е описан в детайли разработеният алгоритъм за намиране на характеристично разпределение. В раздел 4 е дефинирано понятието съкратено характеристично разпределение и е дадена връзката му с тегловното разпределение. Съкратеното характеристично разпределение се явява обобщение на спектъра на Уолш. В раздел 5 е изчислена сложността на предложения алгоритъм и са представени

експериментални резултати.

В **Глава 3** са разгледани методи за намиране на тегловно разпределение, когато линейният код е над съставно крайно поле. В раздел 1 чрез код на следите, получен от разширена пораждаща матрица, задачата е сведена до линеен код над просто поле. В останалите раздели за основа се ползва трансформацията на следите [9]. В раздел 2 е коментиран стандартният подход чрез трансформацията на следите от разширения характеристичен вектор. Представен е по-ефективен алгоритъм при лексикографска наредба на елементите на полето спрямо самодуален базис. В раздели 3 и 4 е описан подобрен алгоритъм за пресмятане на трансформацията на следите, който ползва характеристичен вектор относно симплекс кода. В раздел 3 подобреният алгоритъм е обоснован аналитично чрез матрици, а в раздел 4 е дадено подробното описание и е изчислена сложността.

В **Глава 4** са описани методи за намиране на радиус на покритие на линеен код чрез дискретни трансформации. Обобщено и подобрено е предложеното от Карповски решение за двоични кодове [52]. Дефинирано е понятието съкратено разпределение на вектор, което е вариант на обобщение на трансформацията на Уолш-Адамар. В раздел 1 вниманието е насочено към линейни кодове над прости крайни полета, като е приложена трансформацията на Виленкин-Крестенсон. В раздел 2 са описани резултатите за съставни крайни полета, като е приложена трансформацията на следите.

След всяка глава в кратък коментарен раздел е дадена информация за съавторство, къде са докладвани и публикувани описаните резултати.

В Заключението са обобщени получените резултати.

В Приложение са описани приносните моменти на дисертационния труд, дадени са списъци на изнесените доклади, публикациите по темата на дисертацията и ползваната литература.

Благодарности

Благодаря на моите учители и наставници през годините Стефка Тодорова, Емил Петров, проф. дмн Димитър Вакарелов и доц. д-р Димитър Петров! Благодаря на научния ми ръководител проф. дмн Илия Буюклиев, който прояви много търпение и постоянство, вярваше безрезервно във възможностите ми и ме доведе до завършека на този труд! Благодаря на проф. дмн Стефка Буюклиева за подкрепата и многото труд като съавтор! Благодаря на Тадзя Марута, че ме прие за съавтор! Благодаря на директорите на Института по математика и информатика акад. Юлиан Ревалски, акад. Веселин Дренски и

проф. дмн Петър Бойваленков за подкрепата и доверието, което ми гласуваха! Благодаря на колегите от секция „Математически основи на информатиката“, на ръководителите проф. дмн Емил Колев и доц. д-р Христо Костадинов за подкрепата през цялото време на работата ми по докторантурата! Благодаря за подкрепата и конструктивните разговори на участниците в Националния семинар по теория на кодирането „Професор Стефан Додунеков“ и на колегите от Факултет „Математика и информатика“ на Великотърновския университет „Св. св. Кирил и Методий“!

Глава 1

Основни понятия

1.1 Крайни полета

Нека $\mathbb{F}_q$ е крайно поле с q елемента и характеристика простото число p. Теорията на крайните полета е систематически изградена например в [56, 60].

Всяко крайно разширение F на полето K може да се разглежда като линейно пространство над K. *Базис* на полето F над K се нарича всеки базис на това линейно пространство. В частност, полето $\mathbb{F}_{p^m}$, като крайно разширение от степен m на простото му подполе $\mathbb{F}_p$, е изоморфно на линейното пространство $\mathbb{F}_p^m$. Ако $\beta_1, \ldots, \beta_m \in \mathbb{F}_{p^m}$ образуват базис на $\mathbb{F}_{p^m}$ над $\mathbb{F}_p$, то всеки елемент $\alpha \in \mathbb{F}_{p^m}$ се представя еднозначно като линейна комбинация $\alpha = \lambda_1\beta_1 + \cdots + \lambda_m\beta_m$, където $\lambda_1, \ldots, \lambda_m \in \mathbb{F}_p$.

Дефиниция 1.1. Нека $K = \mathbb{F}_q$, $F = \mathbb{F}_{q^m}$ и $\alpha \in F$. *Следата* $\mathrm{Tr}_{F/K}(\alpha)$ на елемента α над K се определя от равенството

$$\mathrm{Tr}_{F/K}(\alpha) = \alpha + \alpha^q + \alpha^{q^2} + \ldots + \alpha^{q^{m-1}}.$$

Ако K е просто подполе на F, то $\mathrm{Tr}_{F/K}(\alpha)$ се нарича *абсолютна следа* на елемента α и се означава с $\mathrm{Tr}_F(\alpha)$.

В разработката се ползва само абсолютна следа, която за краткост ще бъде наричана *следа*. Когато полето се подразбира, следата ще бъде означавана с $\mathrm{Tr}(\alpha)$.

Някои свойства на следата са:

1. $\mathrm{Tr}_{F/K}(\alpha) \in K$ за всяко $\alpha \in F$;
2. $\mathrm{Tr}_{F/K}(\alpha_1 + \alpha_2) = \mathrm{Tr}_{F/K}(\alpha_1) + \mathrm{Tr}_{F/K}(\alpha_2)$ за всеки $\alpha_1, \alpha_2 \in F$;

3. $\mathrm{Tr}_{F/K}(\lambda\alpha) = \lambda\mathrm{Tr}_{F/K}(\alpha)$ за всеки $\lambda \in K$ и $\alpha \in F$;

4. $\mathrm{Tr}_{F/K}$ е линейно изображение от F върху K, където F и K се разглеждат като линейни пространства над полето K;

5. $\mathrm{Tr}_{F/K}(\alpha) = m\alpha$ за всяко $\alpha \in K$;

6. $\mathrm{Tr}_{F/K}(\alpha^q) = \mathrm{Tr}_{F/K}(\alpha)$ за всяко $\alpha \in F$.

Дефиниция 1.2. Два базиса $\beta_1, \ldots, \beta_m \in F$ и $\beta'_1, \ldots, \beta'_m \in F$ на полето F над полето K се наричат *дуални*, ако за $1 \leq i, j \leq m$ е изпълнено

$$\mathrm{Tr}_{F/K}(\beta_i\beta'_j) = \begin{cases} 0, & \text{при} \quad i \neq j, \\ 1, & \text{при} \quad i = j. \end{cases}$$

Базис, дуален на себе си, се нарича *самодуален*.

За всеки базис съществува еднозначно определен дуален базис.

Теорема 1.1 ([64]). *Съществува самодуален базис на полето $F = \mathbb{F}_{q^m}$ над $K = \mathbb{F}_q$ тогава и само тогава, когато q е четно или q и m са едновременно нечетни.*

За целите на разработката елементите на полето $\mathbb{F}_q$ са линейно наредени и означени съответно с $\alpha_0 = 0, \alpha_1, \ldots, \alpha_{q-1}$. Счита се, че наредбата е фиксирана от лексикографската наредба спрямо фиксиран базис на полето над простото му подполе. Така позицията на нулевия елемент е постоянна, но всеки друг елемент, включително единичният, може да бъде на различна позиция в зависимост от избрания базис.

Например, за полето $\mathbb{F}_4$, при базис $\beta, 1$ се получава наредбата $0, 1, \beta, \beta+1$, а при базис $\beta, \beta+1$ се получава наредбата $0, \beta+1, \beta, 1$.

За простите полета се използва базисът $\alpha_1 = 1$ и естествената наредба в $\mathbb{F}_q \cong \mathbb{Z}_q$, именно $0, 1, \ldots, q-1$. Когато във формула елемент на полето се подлага на операция над множеството $\mathbb{Z}$ на целите числа или над множеството $\mathbb{C}$ на комплексните числа, се счита, че е използван естественият хомоморфизъм от $\mathbb{Z}_q$ в $\mathbb{Z}$ (или $\mathbb{C}$).

Ако $x \in \mathbb{F}_q^k$, координатите на x се означават с долен индекс, т. е. $x = (x_1, x_2, \ldots, x_k)$. Ако два вектора x и x' са от линейното пространство $\mathbb{F}_q^k$, тяхното *Евклидово скаларно произведение* се дефинира с равенството $\langle x, x' \rangle = x_1x'_1 + x_2x'_2 + \cdots + x_kx'_k$, като действията се извършват в полето $\mathbb{F}_q$. Тъй като в разработката се ползва само Евклидово скаларно произведение, то ще бъде наричано *скаларно произведение*.

1.2 Линейни кодове

Дефиниция 1.3. Всяко k-мерно линейно подпространство C на линейното пространство $\mathbb{F}_q^n$ се нарича *q-ичен линеен $[n,k]$ код* (или *линеен $[n,k]_q$ код*). Параметрите n и k се наричат съответно *дължина* и *размерност* на C, а векторите от C се наричат *кодови думи*.

Дефиниция 1.4. *Тегло (по Хеминг)* $\mathrm{wt}(x)$ на вектора $x \in \mathbb{F}_q^n$ е броят на ненулевите му координати.

Дефиниция 1.5. За даден линеен $[n,k]_q$ код C, най-малкото ненулево тегло на кодова дума се нарича *минимално тегло* на кода C и се означава с d. Ако A_w е броят на кодовите думи с дължина w в C, $w = 0, 1, \ldots, n$, то редицата $(A_0, A_1, \ldots, A_n)$ се нарича *тегловно разпределение* на C, а полиномът $W(z) = \sum_{w=0}^{n} A_w z^w$ се нарича *тегловна функция* на кода C.

Дефиниция 1.6 ([57], p. 142)**.** *Пълната тегловна функция* на линейния $[n,k]_q$ код C е полином на q променливи, който се дефинира по следния начин:

$$W_C(z_0, z_1, \ldots, z_{q-1}) = \sum A(t) z_0^{t_0} z_1^{t_1} \cdots z_{q-1}^{t_{q-1}} = \sum_{c \in C} z_0^{s_0} z_1^{s_1} \cdots z_{q-1}^{s_{q-1}}, \tag{1.1}$$

където $s_j = s_j(c)$ е броят на координатите на кодовата дума c, равни на $\alpha_j \in \mathbb{F}_q$, $j = 0, 1, \ldots, q-1$, а $A(t)$ е броят на кодовите думи $c \in C$, за които $(s_0, s_1, \ldots, s_{q-1}) = t = (t_0, t_1, \ldots, t_{q-1}) \in \mathbb{Z}^q$.

Дефиниция 1.7. Всяка $k \times n$ матрица G, чиито редове формират базис на линейния $[n,k]_q$ код C, се нарича *пораждаща матрица* на кода C.

Дефиниция 1.8. Матрица H с размерност $(n-k) \times n$, която определя код C в следния смисъл

$$C = \{x \in \mathbb{F}_q^n | Hx^{\mathrm{T}} = 0\},$$

се нарича *проверочна матрица* на кода C.

Редовете на H са линейно независими.

Теорема 1.2 ([49], Theorem 1.4.13)**.** *Нека C е линеен $[n,k]_q$ код с проверочна матрица H. Ако $c \in C$, стълбовете на H, съответстващи на ненулевите координати на c, са линейно зависими. Обратно, ако между w стълба на H има линейна зависимост с ненулеви коефициенти, то съществува кодова дума в C с тегло w, чиито ненулеви координати съответстват на тези стълбове.*

Всеки $d-1$ стълба на проверочната матрица са линейно независими.

Дефиниция 1.9. Нека H е проверочна матрица на линейния $[n,k]_q$ код C. Линейният $[n,n-k]_q$ код с пораждаща матрица H се нарича *дуален* на C и се означава с $C^{\perp}$.

Дефиниция 1.10. За линеен $[n,k]_q$ код C и произволен вектор $x \in \mathbb{F}_q^n$ множеството $x+C = \{x+c | c \in C\}$ се нарича *съседен клас* на кода C. *Тегло на съседен клас* е най-малкото тегло на вектор от съседния клас, а произволен вектор с това най-малко тегло от съседния клас се нарича *лидер на съседния клас*.

Нулевият вектор е единствен лидер на самия код C като съседен клас.

Дефиниция 1.11. *Синдром* на вектора $x \in \mathbb{F}_q^n$ по отношение на проверочната матрица H на даден линеен $[n,k]_q$ код C е векторът $syn(x) = Hx^{\mathrm{T}} \in \mathbb{F}_q^{n-k}$.

Теорема 1.3 ([49], Theorem 1.11.5)**.** *Два вектора принадлежат на един и същ съседен клас тогава и само тогава, когато имат един и същ синдром.*

Съседните класове на линеен $[n,k]_q$ код C са непресичащи се и обединението им дава цялото множество $\mathbb{F}_q^n$. Всеки съседен клас съдържа q^k елемента, а броят на съседните класове е q^{n-k}.

Дефиниция 1.12. Максималното измежду теглата на съседните класове на линейния $[n,k]_q$ код C се нарича *радиус на покритие* на C и се означава с $R(C)$.

Теорема 1.4 ([49], Theorem 1.12.5)**.** *$R(C)$ е най-малкото число s такова, че всеки ненулев синдром е линейна комбинация на s или по-малко стълбове от проверочната матрица H, а някой синдром изисква s стълба.*

Лема 1.5 ([47])**.** *Нека C е линеен $[n,k]_q$ код с проверочна матрица $H = (h_1 h_2 \ldots h_n)$, където h_i, $i = 1, \ldots, n$, са стълбовете на матрицата H. Тогава:*

1. *Теглото на лидера на съседния клас $x+C$ е равно на най-малкото цяло число l, такова че $Hx^{\mathrm{T}} = a_1 h_{t_1} + a_2 h_{t_2} + \ldots + a_l h_{t_l}$ за някои $a_i \in \mathbb{F}_q$ и $h_{t_i} \in \{h_1, h_2, \ldots, h_n\}$ за $1 \leq i \leq l$.*

2. *Радиусът на покритие на кода C е най-малкото естествено число r, такова че всеки ненулев вектор-стълб с $n-k$ координати е линейна комбинация на не повече от r стълба на H.*

Дефиниция 1.13. *Линеен код с пълна дължина* е линеен код без нулеви стълбове в пораждащата матрица.

Ако C е линеен $[n,k]_q$ код с пълна дължина, а C' е получен от C с добавяне на s нулеви стълба в пораждащата матрица, то C' е линеен код с дължина $n+s$, същата размерност k, същото тегловно разпределение като C и радиус на покритие $R(C') = R(C) + s$. Следователно, достатъчно е да се пресметнат тегловното разпределение и радиуса на покритие на кода C, за да станат ясни съответните параметри на кода C'.

Една матрица се нарича *мономиална*, ако има точно един ненулев елемент във всеки ред и стълб. Всяка мономиална матрица е квадратна и може да се представи като произведение на диагонална и пермутационна матрица.

Дефиниция 1.14 ([49])**.** Нека C и C' са линейни $[n,k]_q$ кодове и G е пораждаща матрица на C. Тогава C и C' се наричат *мономиално еквивалентни*, ако съществува мономиална матрица A, така че $G \cdot A$ е пораждаща матрица на C'.

Линейните $[n,k]_q$ кодове C и C' са мономиално еквивалентни точно тогава, когато съществува *мономиално изображение*, представено от мономиалната матрица A, при което $C' = \{cA | c \in C\}$.

Теорема 1.6 ([18])**.** *Два линейни $[n,k]_q$ кода са мономиално еквивалентни тогава и само тогава, когато съществува линеен изоморфизъм между тях, който запазва теглата.*

Максималният брой от по двойки линейно независими вектори в линейното пространство $\mathbb{F}_q^k$ е $\theta(q,k) = \frac{q^k-1}{q-1}$. Това е броят на едномерните линейни подпространства на $\mathbb{F}_q^k$.

Дефиниция 1.15. Матрица с размерност $k \times \theta(q,k)$, чиито стълбове са по двойки линейно независими вектори от $\mathbb{F}_q^k$, поражда линеен $[\theta(q,k),k]_q$ код, който се нарича *симплекс код* и се означава с $\mathcal{S}_{q,k}$.

Две $k \times \theta(q,k)$ матрици с това свойство (чиито стълбове са по двойки линейно независими вектори от $\mathbb{F}_q^k$) пораждат мономиално еквивалентни линейни кодове. Затова симплекс кодовете се означават по един и същ начин с $\mathcal{S}_{q,k}$.

1.3 Дискретна трансформация на Уолш-Адамар

Функция $f : \mathbb{F}_2^k \to \mathbb{F}_2$ се нарича *булева функция на n променливи*. Векторът от стойностите на функцията f при лексикографска наредба на аргументите се нарича *таблица на истинност* на функцията f и се означава с TT_f. Векторът TT_f е с дължина 2^k.

Дефиниция 1.16 ([51]). Нека f е булева функция на k променливи. *Дискретна трансформация на Уолш-Адамар* на f е функцията $\widehat{f} : \mathbb{F}_2^k \to \mathbb{Z}$, дефинирана с равенството

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_2^k} f(x)(-1)^{\langle x,\omega \rangle}, \quad \omega \in \mathbb{F}_2^k. \tag{1.2}$$

Таблицата на истинност на функцията $\widehat{f}$ се нарича *Уолш спектър* на функцията f и се означава с W_f.

Трансформационните матрици се дефинират индуктивно, както следва:

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}, \; k > 1. \tag{1.3}$$

Теорема 1.7. *Ако f е булева функция на k променливи, то $W_f = H_k \cdot TT_f$.*

Доказателство. Индукция по k. При $k = 1$, нека $f : \mathbb{F}_2 \to \mathbb{F}_2$ е булева функция на една променлива. Тогава $TT_f = \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}$ и

$$W_f(0) = f(0).(-1)^{0.0} + f(1).(-1)^{1.0} = f(0) + f(1),$$

$$W_f(1) = f(0).(-1)^{0.1} + f(1).(-1)^{1.1} = f(0) - f(1).$$

Очевидно $W_f = H_1 \cdot TT_f$.

Нека $k \geq 1$ и $f : \mathbb{F}_2^{k+1} \to \mathbb{F}_2$ е булева функция на $k+1$ променливи с таблица на истинност TT_f. Нека булевите функции на k променливи f_{x_0} са такива, че $f_{x_0}(x_1, \ldots, x_k) = f(x_0, x_1, \ldots, x_k)$, $x_0 = 0, 1$. При лексикографската наредба векторите $(x_0, x_1, \ldots, x_k) \in \mathbb{F}_2^{k+1}$ могат да бъдат разделени на две групи от по 2^k вектора – първата при $x_0 = 0$ и втората при $x_0 = 1$. Така TT_f може да се раздели на два 2^k-мерни вектора TT_{f_0} и TT_{f_1}. Съгласно индукционното предположение $W_{f_0} = H_k \cdot TT_{f_0}$ и $W_{f_1} = H_k \cdot TT_{f_1}$. Следователно

$$\begin{aligned} W_f(\omega) &= \sum_{x \in \mathbb{F}_2^{k+1}} f(x)(-1)^{\langle x,\omega \rangle} \\ &= \sum_{(x_1,\ldots,x_k) \in \mathbb{F}_2^k} f(0, x_1, \ldots, x_k)(-1)^{\langle (0,x_1,\ldots,x_k),(\omega_0,\omega_1,\ldots,\omega_k) \rangle} \\ &\quad + \sum_{(x_1,\ldots,x_k) \in \mathbb{F}_2^k} f(1, x_1, \ldots, x_k)(-1)^{\langle (1,x_1,\ldots,x_k),(\omega_0,\omega_1,\ldots,\omega_k) \rangle} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{(x_1,\ldots,x_k)\in\mathbb{F}_2^k} f_0(x_1,\ldots,x_k)(-1)^{\langle(x_1,\ldots,x_k),(\omega_1,\ldots,\omega_k)\rangle} \\
&\quad + \sum_{(x_1,\ldots,x_k)\in\mathbb{F}_2^k} f_1(x_1,\ldots,x_k)(-1)^{\omega_0+\langle(x_1,\ldots,x_k),(\omega_1,\ldots,\omega_k)\rangle} \\
&= \sum_{(x_1,\ldots,x_k)\in\mathbb{F}_2^k} f_0(x_1,\ldots,x_k)(-1)^{\langle(x_1,\ldots,x_k),(\omega_1,\ldots,\omega_k)\rangle} \\
&\quad +(-1)^{\omega_0} \sum_{(x_1,\ldots,x_k)\in\mathbb{F}_2^k} f_1(x_1,\ldots,x_k)(-1)^{\langle(x_1,\ldots,x_k),(\omega_1,\ldots,\omega_k)\rangle}.
\end{aligned}
$$

Тогава

$$
\begin{aligned}
W_f(\omega) &= W_{f_0}(\omega_1,\ldots,\omega_k) + W_{f_1}(\omega_1,\ldots,\omega_k) \quad \text{при} \quad \omega_0 = 0, \\
W_f(\omega) &= W_{f_0}(\omega_1,\ldots,\omega_k) - W_{f_1}(\omega_1,\ldots,\omega_k) \quad \text{при} \quad \omega_0 = 1.
\end{aligned}
$$

Последните равенства и индукционното предположение показват, че

$$
\begin{aligned}
W_f &= \begin{pmatrix} W_{f_0} + W_{f_1} \\ W_{f_0} - W_{f_1} \end{pmatrix} = \begin{pmatrix} H_k \cdot TT_{f_0} + H_k \cdot TT_{f_1} \\ H_k \cdot TT_{f_0} - H_k \cdot TT_{f_1} \end{pmatrix} \\
&= \begin{pmatrix} H_k & H_k \\ H_k & -H_k \end{pmatrix} \cdot \begin{pmatrix} TT_{f_0} \\ TT_{f_1} \end{pmatrix}.
\end{aligned}
$$

С това индукционната стъпка е доказана. □

Матриците H_k са именувани на Силвестър и Адамар заради работата им с подобни матрици.

Дефиниция 1.17 ([57])**.** Матрицата M от ред t се нарича *Адамарова от ред* t, ако е с размерност $t \times t$, елементите ѝ са 1 и -1, при което е изпълнено $MM^{\mathrm{T}} = tI$, където I е единичната матрица от ред t.

Различните редове (стълбове) на Адамарова матрица са ортогонални, а скаларният квадрат на даден ред (стълб) е равен на дължината му. Обратната матрица е $M^{-1} = \frac{1}{t}M$. Адамар [45] показва, че Адамаровите матрици имат максимална детерминанта измежду $t \times t$ матриците с елементи в интервала $[-1, 1]$. Конструкцията (1.3) е открита от Силвестър [69].

Съвсем естествено, дефиниция 1.16 може да се обобщи за псевдобулеви функции, т. е. функции от вида $f : \mathbb{F}_q^k \to \mathbb{Z}$, при което

$$
\widehat{f}(\omega) = \sum_{x\in\mathbb{F}_2^k} f(x)(-1)^{\langle x,\omega\rangle}, \quad \omega \in \mathbb{F}_2^k. \tag{1.4}
$$

Нека G е пораждаща матрица на линеен $[n,k]_2$ код, а $f : \mathbb{F}_2^k \to \mathbb{Z}$ е характеристична функция в смисъл, че $f(x)$ е броят на стълбовете в G, които са равни на x. В този случай трансформацията на Уолш-Адамар $\widehat{f}$ кореспондира с теглото на кодовата дума ωG по следния начин

$$\mathrm{wt}(\omega G) = \frac{n - \widehat{f}(\omega)}{2}, \quad \omega \in \mathbb{F}_2^k. \tag{1.5}$$

Този факт е споменат от Карповски [51] в случая, когато в G няма нулеви или повтарящи се стълбове, т. е. когато дуалният код е с минимално тегло, по-голямо от 2.

Наистина, кодовите думи на C са линейни комбинации от редовете на пораждащата матрица G, т. е. могат да бъдат получени, като произволен вектор ред $\omega \in \mathbb{F}_2^k$ се умножи по матрицата G. Дадена координата на вектора ωG е скаларно произведение на ω и съответния стълб от G. Теглото $\mathrm{wt}(\omega G)$ е броят на единиците в ωG. Тогава броят на нулите в ωG е $n - \mathrm{wt}(\omega G)$. От друга страна, събираемите в (1.4) са различни от 0, точно когато x е стълб в G. За всеки стълб x на матрицата G

$$(-1)^{\langle x,\omega\rangle} = \begin{cases} 1, & \text{ако } \langle x,\omega\rangle = 0, \\ -1, & \text{ако } \langle x,\omega\rangle = 1. \end{cases} \tag{1.6}$$

Тъй като $f(x)$ е броят на равните на x стълбове в G, то сумата в (1.4) може да се представи като сума на изрази от вида (1.6) за всеки стълб от G. Тогава

$$\widehat{f}(\omega) = n - \mathrm{wt}(\omega G) - \mathrm{wt}(\omega G),$$

откъдето се извежда (1.5).

1.4 Кронекерова степен. Бързи трансформации

Кронекерово произведение на матриците $A = (a_{ij})_{s_1 \times t_1}$ и $B = (b_{ij})_{s_2 \times t_2}$ е $s_1 s_2 \times t_1 t_2$ матрицата

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1t_1}B \\ a_{21}B & a_{22}B & \dots & a_{2t_1}B \\ \dots & & & \\ a_{s_1 1}B & a_{s_1 2}B & \dots & a_{s_1 t_1}B \end{pmatrix}.$$

Елементарни свойства на Кронекеровото произведение са:

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C, \tag{1.7}$$

$$(A+B)\otimes(C+D)=A\otimes C+A\otimes D+B\otimes C+B\otimes D, \tag{1.8}$$

$$(A\otimes B)(C\otimes D)=(AC)\otimes(BD). \tag{1.9}$$

Кронекеровото произведение не е комутативно.

За квадратната матрица M се дефинира k-та Кронекерова степен $\otimes^k M$ чрез рекурентните формули:

$$\otimes^2 M=M\otimes M,\quad \otimes^{k+1}M=M\otimes\left(\otimes^k M\right), k>1.$$

Ако M е квадратна матрица от ред t, то $\otimes^k M$ е от ред t^k.

Гуд [42] показва, че Кронекеровото произведение може да се представи като произведение (в обичайния смисъл) на разредени матрици. Следващата теорема е преформулировка за случая на Кронекерова степен.

Теорема 1.8. *Нека M е квадратна матрица от ред t и k е естествено число. Тогава*

$$\otimes^k M=B_1\cdot B_2\cdots B_k, \tag{1.10}$$

където $B_l=I_{t^{l-1}}\otimes M\otimes I_{t^{k-l}}$, $1\le l\le k$, а I_s е единичната матрица от ред s.

Доказателство. Доказателство по индукция. Нека твърдението е в сила за всички Кронекерови степени на M, по-малки от k. В частност,

$$\otimes^{k-1}M=B'_1\cdot B'_2\cdots B'_{k-1},$$

където $B'_l=I_{t^{l-1}}\otimes M\otimes I_{t^{k-l-1}}$, $l=1,2,\ldots,k-1$. С прилагане на (1.7), (1.9) и индукционното предположение се получава

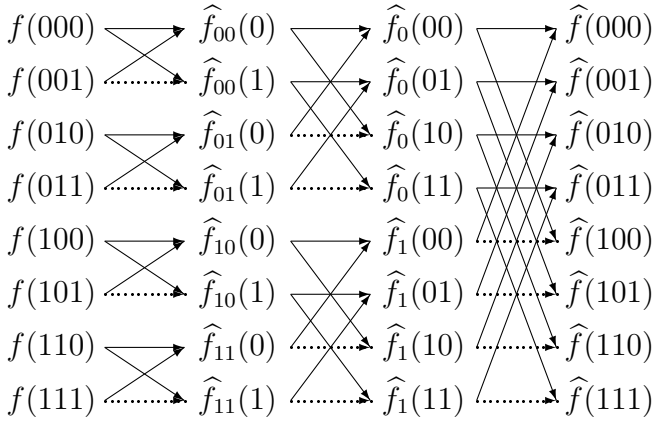
$$\begin{aligned}
\otimes^k M &= M\otimes\left(\otimes^{k-1}M\right)=(M\cdot I_t)\otimes\left(I_{t^{k-1}}\cdot\left(\otimes^{k-1}M\right)\right)\\
&= (M\otimes I_{t^{k-1}})\cdot\left(I_t\otimes\left(\otimes^{k-1}M\right)\right)\\
&= B_1\cdot(I_t^{k-1}\otimes(B'_1\cdot B'_2\cdots B'_{k-1}))\\
&= B_1\cdot(I_t\otimes B'_1)\cdot(I_t\otimes B'_2)\cdots(I_t\otimes B'_{k-1})\\
&= B_1\cdot(I_t\otimes M\otimes I_{t^{k-2}})\cdot(I_t\otimes I_t\otimes M\otimes I_{t^{k-3}})\cdots(I_t\otimes I_{t^{k-2}}\otimes M)\\
&= B_1\cdot B_2\cdot B_3\cdots B_k.
\end{aligned}$$

Оттук следва верността на твърдението за всяко естествено k. □

Пример 1.1. Нека $M=H_1$. Сравнението с (1.3) показва, че $\otimes^k H_1=H_k$. За $k=3$ е изпълнено $H_3=(H_1\otimes I_4)\cdot(I_2\otimes H_1\otimes I_2)\cdot(I_4\otimes H_1)$.

$$H_3=\begin{pmatrix} I_4 & I_4\\ I_4 & -I_4\end{pmatrix}\cdot\begin{pmatrix} I_2 & I_2 & & \\ I_2 & -I_2 & & \\ & & I_2 & I_2\\ & & I_2 & -I_2\end{pmatrix}\cdot\begin{pmatrix} H_1 & & & \\ & H_1 & & \\ & & H_1 & \\ & & & H_1\end{pmatrix}$$

Всеки ред от матриците вдясно съдържа точно два ненулеви елемента, именно 1 или -1. Така, умножението на ред от тези матрици по вектор може да се направи само с една аритметична операция (събиране или изваждане). Процесът може да се представи със следващата диаграма на бърз алгоритъм за умножението $H_3 \cdot TT_f$ при означенията от доказателството на теорема 1.7. В диаграмата непрекъснатите линии означават събиране, а точковите линии – изваждане.



Лехнер [55] прилага (1.10) за трансформацията на Уолш-Адамар, но в обратен ред на множителите, като споменава, че множителите комутират. Това е твърдението на следващата теорема.

Теорема 1.9. *Множителите в* (1.10) *комутират. Така няма значение редът на умножение.*

Доказателство. Прилагат се свойствата на Кронекеровото произведение.

$$\begin{aligned}
B_i \cdot B_j &= (I_{t^{i-1}} \otimes M \otimes I_{t^{n-i}}) \cdot (I_{t^{j-1}} \otimes M \otimes I_{t^{k-j}}) \\
&= I_{t^{i-1}} \otimes ((M \otimes I_{t^{k-i}}) \cdot (I_{t^{j-i}} \otimes M \otimes I_{t^{k-j}})) \\
&= I_{t^{i-1}} \otimes ((M \otimes I_{t^{j-i}}) \cdot (I_{t^{j-i}} \otimes M)) \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes ((M \otimes I_{t^{j-i}}) \cdot (I_t \otimes I_{t^{j-i-1}} \otimes M)) \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes (M \cdot I_t) \otimes (I_{t^{j-i}} \cdot (I_{t^{j-i-1}} \otimes M)) \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes M \otimes I_{t^{j-i-1}} \otimes M \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes (I_t \cdot M) \otimes ((I_{t^{j-i-1}} \otimes M) \cdot I_{t^{j-i}}) \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes ((I_t \otimes I_{t^{j-i-1}} \otimes M) \cdot (M \otimes I_{t^{j-i}})) \otimes I_{t^{k-j}} \\
&= I_{t^{i-1}} \otimes ((I_{t^{j-i}} \otimes M) \cdot (M \otimes I_{t^{j-i}})) \otimes I_{t^{k-j}} = B_j \cdot B_i.
\end{aligned}$$

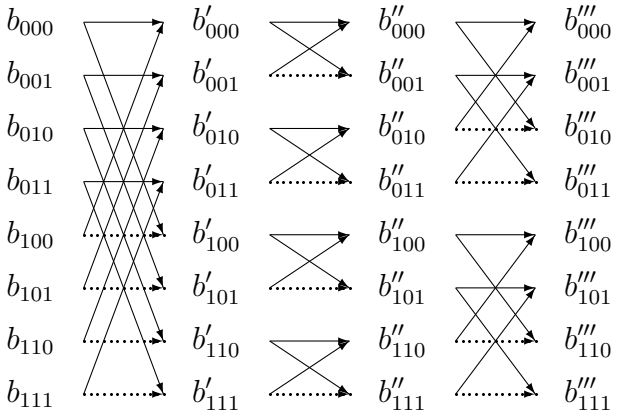
□

Горното твърдение позволява да се разменят итеративните стъпки в алгоритмите за бързо пресмятане.

Пример 1.2.

$$H_3 = (I_4 \otimes H_1) \cdot (H_1 \otimes I_4) \cdot (I_2 \otimes H_1 \otimes I_2).$$

Това води до следващата диаграма на бърз алгоритъм за умножението $H_3 \cdot b$.



1.5 Дискретна трансформация на Виленкин-Крестенсон

Нека ξ е примитивен комплексен q-ти корен на единицата. Матриците на Виленкин-Крестенсон от ред k се дефинират рекурентно, както следва:

$$V_1 = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{q-1} \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(q-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & \xi^{q-1} & \xi^{2(q-1)} & \dots & \xi^{(q-1)^2} \end{pmatrix}, \; V_{k+1} = V_1 \otimes V_k, \; k \in \mathbb{Z}, \, k \geq 1, \quad (1.11)$$

където $\otimes$ означава Кронекерово произведение. Елементите на матрицата V_k са от вида $v_\omega(x) = \xi^{\langle \omega, x \rangle}$, където индексите по редовете и стълбовете, съответно $\omega, x \in \mathbb{Z}_q^k$, са лексикографски наредени. В текста нататък ще бъдат ползвани някои свойства на $v_\omega(x)$, следващи директно от дефиницията, а именно

$$v_\omega(x) = v_x(\omega), \quad v_\omega(x) v_\omega(x') = v_\omega(x + x'), \quad v_\omega(0) = 1. \quad (1.12)$$

Първото свойство показва, че матриците V_k са симетрични.

Дефиниция 1.18. Нека $f : \mathbb{Z}_q^k \to \mathbb{C}$ е функция. *Трансформация на Виленкин-Крестенсон* на f е функцията $\widehat{f} : \mathbb{Z}_q^k \to \mathbb{C}$, дефинирана чрез

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{Z}_q^k} f(x) v_\omega(x), \quad \omega \in \mathbb{Z}_q^k. \tag{1.13}$$

Подробна информация за тази трансформация, както и за други дискретни трансформации, свързани с преобразуването на Фурие, има например в [15, 40, 53].

Нека TT_f е векторът от стойностите на функцията f, когато елементите на $\mathbb{Z}_q^k$ са подредени лексикографски. Това е аналог на таблицата за истинност на булева функция, но тук координатите на $TT_{\widehat{f}}$ са комплексни числа. Векторите от стойностите на функциите f и $\widehat{f}$ са свързани чрез равенството

$$TT_{\widehat{f}} = V_k \cdot TT_f.$$

По този начин трансформацията на Виленкин-Крестенсон се превръща в умножение на матрица с вектор.

Нека q е просто число, G е пораждаща матрица на линеен $[n, k]_q$ код с пълна дължина и функцията $f : \mathbb{F}_q^k \to \mathbb{Z}$ е дефинирана така, че $f(x)$ е броят на стълбовете в G, които са пропорционални (с ненулев коефициент) на x. В този случай трансформацията на Виленкин-Крестенсон $\widehat{f}$ кореспондира с теглото на кодовата дума ωG по следния начин

$$\mathrm{wt}(\omega G) = \frac{(q-1)n - \widehat{f}(\omega)}{q}, \ \ \omega \in \mathbb{F}_q^k. \tag{1.14}$$

Наистина, кодовите думи на C са линейни комбинации от редовете на пораждащата матрица G, т. е. могат да бъдат получени, като произволен вектор ред $\omega \in \mathbb{F}_q^k$ се умножи по матрицата G. Дадена координата на вектора ωG е скаларно произведение на ω и съответния стълб от G. Теглото $\mathrm{wt}(\omega G)$ е броят на ненулевите координати в ωG. Тогава броят на нулите в ωG е $n - \mathrm{wt}(\omega G)$. От друга страна, събираемите в (1.13) са различни от 0, точно когато има пропорционален на x стълб в G. Ако $f(x) \neq 0$, за някое $x \neq 0$, то $f(x) = f(\alpha x)$ за всяко $\alpha \in \mathbb{F}_q \backslash \{0\}$. За всеки стълб x на матрицата G

$$\sum_{\alpha \in \mathbb{F}_q \backslash \{0\}} v_\omega(\alpha x) = \sum_{\alpha \in \mathbb{F}_q \backslash \{0\}} \xi^{\langle \omega, \alpha x \rangle} = \begin{cases} q-1, & \text{ако } \langle \omega, x \rangle = 0, \\ -1, & \text{ако } \langle \omega, x \rangle \neq 0. \end{cases} \tag{1.15}$$

Тъй като кодът е с пълна дължина, то $f(0) = 0$. Събираемите в (1.13) могат да се прегрупират така, че на всеки стълб в G да съответства точно една сума

от вида (1.15). Тогава

$$\widehat{f}(\omega) = (q-1)(n - \mathrm{wt}(\omega G)) - \mathrm{wt}(\omega G),$$

откъдето се извежда (1.14).

1.6 Трансформация на следите

Обобщение на трансформацията на Уолш-Адамар, което се прилага за параметри на линейни кодове над съставни крайни полета, е предложено от Карповски [52]. Той предлага използването на това преобразуване за изчисляване на тегловното разпределение на съседни класове. Същността на алгоритъма е описана в края на раздела.

Нека $\mathbb{F}_q$ съставно крайно поле, като $q = p^m$ и p е просто число.

За трансформацията на следите вместо скаларното произведение в трансформацията на Виленкин-Крестенсон се ползва абсолютната следа на скаларното произведение [9, p. 367].

Нека ζ е примитивен комплексен p-ти корен на 1 и са дефинирани изразите

$$\tau_\omega(x) = \zeta^{\mathrm{Tr}(\langle \omega, x \rangle)} \tag{1.16}$$

за произволни $\omega, x \in \mathbb{F}_q^k$. Както по-рано бе отбелязано, ползва се естественият хомоморфизъм между $\mathbb{F}_p$ и комплексните числа с абсолютна стойност 1. Трансформацията се определя от матрицата $T_k = (\tau_\omega(x))$ с размери $q^k \times q^k$, в която индексите $\omega, x \in \mathbb{F}_q^k$ са наредени лексикографски. Чрез равенството (1.16) се дефинира преобразуване от тип Фурие [9], което се нарича трансформация на следите.

Дефиниция 1.19. Нека $\mathbb{F}_q$ е крайно поле с q елемента, $q = p^m$ за простото число p, а ζ е примитивен комплексен p-ти корен на 1. *Трансформация на следите* на функцията $f : \mathbb{F}_q^k \to \mathbb{C}$ е функцията $\widehat{f} : \mathbb{F}_q^k \to \mathbb{C}$, дефинирана с

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_q^k} f(x)\tau_\omega(x) = \sum_{x \in \mathbb{F}_q^k} f(x)\zeta^{\mathrm{Tr}(\langle \omega, x \rangle)}, \quad \omega \in \mathbb{F}_q^k. \tag{1.17}$$

Векторите от стойностите на функциите f и $\widehat{f}$ са свързани с равенството

$$TT_{\widehat{f}} = T_k \cdot TT_f.$$

От симетричността и линейността на скаларното произведение и следата се получава

$$\tau_\omega(x) = \tau_x(\omega), \quad \tau_\omega(x)\tau_\omega(x') = \tau_\omega(x + x'). \tag{1.18}$$

Лема 1.10. *За $x \in \mathbb{F}_q^k$ е в сила равенството*

$$\sum_{\omega \in \mathbb{F}_q^k} \tau_\omega(x) = \begin{cases} q^k, & \text{за } x = 0, \\ 0, & \text{за } x \neq 0. \end{cases}$$

Доказателство. Тази лема е модификация на [57, Chapter 5, Lemma 9]. Приложеното тук доказателство е за пълнота на изложението. Ползва се индукция по k. В базовата стъпка $k = 1$ е изпълнено

$$\sum_{\omega \in \mathbb{F}_q} \tau_\omega(x) = \sum_{\omega \in \mathbb{F}_q} \zeta^{\mathrm{Tr}(x\omega)} = \begin{cases} q, & \text{за } x = 0, \\ 0, & \text{за } x \neq 0. \end{cases}$$

Следва обосновка за случая $x \neq 0$. Когато ω пробягва $\mathbb{F}_q$, произведението $x\omega$ пробягва всички елементи на $\mathbb{F}_q$. Следата е линейно изображение върху $\mathbb{F}_p$ с ядро от p^{m-1} елемента. Така

$$\sum_{\omega \in \mathbb{F}_q} \zeta^{\mathrm{Tr}(x\omega)} = \sum_{\omega \in \mathbb{F}_q} \zeta^{\mathrm{Tr}(\omega)} = p^{m-1} \sum_{\omega \in \mathbb{F}_p} \zeta^\omega = p^{m-1} \frac{1 - \zeta^p}{1 - \zeta} = 0.$$

Нека равенството е изпълнено за някое естествено число k и е даден векторът $x = (x_0, x') \in \mathbb{F}_q^{k+1}$, $x_0 \in \mathbb{F}_q$, $x' \in \mathbb{F}_q^k$. Поради линейността на следата като изображение, се получава

$$\begin{aligned} \sum_{\omega \in \mathbb{F}_q^{k+1}} \tau_\omega(x) &= \sum_{\omega \in \mathbb{F}_q^{k+1}} \zeta^{\mathrm{Tr}(\langle \omega, x \rangle)} = \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\mathrm{Tr}(jx_0 + \langle \omega', x' \rangle)} \\ &= \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\mathrm{Tr}(jx_0) + \mathrm{Tr}(\langle \omega', x' \rangle)} = \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\mathrm{Tr}(jx_0)} \zeta^{\mathrm{Tr}(\langle \omega', x' \rangle)} \\ &= \left(\sum_{j \in \mathbb{F}_q} \zeta^{\mathrm{Tr}(jx_0)} \right) \left(\sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\mathrm{Tr}(\langle \omega', x' \rangle)} \right) \\ &= \left(\sum_{j \in \mathbb{F}_q} \tau_j(x_0) \right) \left(\sum_{\omega' \in \mathbb{F}_q^k} \tau_{\omega'}(x') \right) \end{aligned} \tag{1.19}$$

Съгласно индукционното предположение и базовата стъпка горната сума е ненулева точно тогава, когато $x_0 = 0$ и $x' = 0$. В този случай сумата е $q.q^k = q^{k+1}$.

Сега съгласно принципа на математическата индукция може да се заключи, че равенството е изпълнено за всяко естествено k. □

При извеждане на равенство (1.19) е ползвано равенството

$$\zeta^{\mathrm{Tr}(\langle\omega,x\rangle)} = \zeta^{\mathrm{Tr}(jx_0)}\zeta^{\mathrm{Tr}(\langle\omega',x'\rangle)},$$

където $\omega = (j, \omega'), x = (x_0, x') \in \mathbb{F}_q^{k+1}$, $j, x_0 \in \mathbb{F}_q$, $\omega', x' \in \mathbb{F}_q^k$. Последното показва, че матриците T_k са свързани чрез Кронекерово произведение, т. е. $T_{k+1} = T_1 \otimes T_k$ и $T_k = \otimes^k T_1$ за $k \in \mathbb{N}$.

Нека G е пораждаща матрица на линеен $[n,k]_q$ код C с пълна дължина.

Дефиниция 1.20. Стойността на *характеристичната функция* $f_G(x)$ на линейния $[n,k]_q$ код C с пораждаща матрица G е броят на стълбовете на G, които са пропорционални (с ненулев коефициент) на x, за $x \in \mathbb{F}_q^k$.

За кодове с пълна дължина $f_G(0) = 0$.

Един линеен код може да има различни характеристични функции, тъй като стойностите зависят от избраната пораждаща матрица. Ако пораждащата матрица е ясна от контекста, е допустимо да се записва само $f(x)$.

Забележка 1.1. Карповски [52] разглежда разширената матрица

$$G' = (\alpha_1 G | \alpha_2 G | \ldots | \alpha_{q-1} G) \tag{1.20}$$

при условие, че в G няма пропорционални стълбове, т. е. минималното тегло на дуалния код е по-голямо от 2. В този случай характеристичната функция в класическия смисъл $f' : \mathbb{F}_q^k \to \mathbb{F}_2$, която за всеки вектор x показва дали е стълб в G', съвпада с дефинираната по-горе функция f_G.

Теорема 1.11. *Нека G е пораждаща матрица на линейния $[n,k]_q$ код с пълна дължина C. Тогава за теглата на кодовите думи на C е изпълнено*

$$\mathrm{wt}(\omega G) = \frac{(q-1)n - \widehat{f}(\omega)}{q}, \quad \omega \in \mathbb{F}_q^k, \tag{1.21}$$

където $\widehat{f}$ е трансформацията на следите на характеристичната функция f_G на кода C.

Доказателство. Кодовите думи на C са линейни комбинации от редовете на пораждащата матрица G, т. е. могат да бъдат получени, като произволен вектор ред $\omega \in \mathbb{F}_q^k$ се умножи по матрицата G. Дадена координата на вектора ωG е скаларно произведение на ω и съответния стълб от G. Теглото $\mathrm{wt}(\omega G)$ е броят на ненулевите координати в ωG. Тогава броят на нулите в ωG е $n - \mathrm{wt}(\omega G)$.

От друга страна, събираемите в (1.17) са различни от 0, точно когато има пропорционален на x стълб в G. Ако $f_G(x) \neq 0$, за някое $x \neq 0$, то $f_G(x) = f_G(\alpha x)$ за всяко $\alpha \in \mathbb{F}_q \backslash \{0\}$. За всеки стълб x на матрицата G е изпълнено

$$\sum_{\alpha \in \mathbb{F}_q \backslash \{0\}} \tau_\omega(\alpha x) = \sum_{\alpha \in \mathbb{F}_q \backslash \{0\}} \zeta^{\mathrm{Tr}(\langle \omega, \alpha x \rangle)} = \begin{cases} q-1, & \text{ако } \langle \omega, x \rangle = 0, \\ -1, & \text{ако } \langle \omega, x \rangle \neq 0. \end{cases} \tag{1.22}$$

Равенството в първата хипотеза е очевидно. Нека $\langle \omega, x \rangle \neq 0$. Тогава $\{\alpha \langle \omega, x \rangle \,|\, \alpha \in \mathbb{F}_q, \alpha \neq 0\} = \mathbb{F}_q \backslash \{0\}$. Абсолютната следа Tr е линейно изображение върху $\mathbb{F}_p$ с ядро от p^{m-1} елемента. Следователно в сумата (1.22) ще има точно $p^{m-1} - 1$ събираеми $\zeta^0 = 1$ и по p^{m-1} събираеми ζ^λ за всяко $\lambda \in \mathbb{F}_p$, $\lambda \neq 0$. Оттук

$$\sum_{\alpha \in \mathbb{F}_q \backslash \{0\}} \zeta^{\mathrm{Tr}(\langle \omega, \alpha x \rangle)} = p^{m-1} - 1 + p^{m-1} \sum_{\lambda \in \mathbb{F}_p \backslash \{0\}} \zeta^\lambda = -1 + p^{m-1} \sum_{\lambda \in \mathbb{Z}_p} \zeta^\lambda = -1.$$

Тъй като кодът е с пълна дължина, то $f_G(0) = 0$. Събираемите в (1.17) могат да се прегрупират така, че на всеки стълб в G да съответства точно една сума от вида (1.22). Тогава

$$\widehat{f}(\omega) = (q-1)(n - \mathrm{wt}(\omega G)) - \mathrm{wt}(\omega G),$$

откъдето се извежда (1.21). □

Коментари към Глава 1

Интересът към прилагане на трансформацията на Уолш-Адамар на колегията идва от заниманията на Душан Биков като докторант на проф. Стефка Буюклиева и общата им работа с проф. Илия Буюклиев [24]. Те прилагат трансформацията към поляризираната таблица за истинност на булева функция. От получения спектър се установява нелинейност на изследваната функция.

Основен принос на автора при съставяне на тази глава е проучването и систематизирането на информацията за дискретните трансформации на Уолш-Адамар, Виленкин-Крестенсон и следите, като е показано приложението им за намиране на тегловно разпределение на линеен код.

Описаните в тази глава резултати са докладвани [D2, D3, D6, D7, D8, D9, D12, D13] и публикувани в статиите [P1, P2, P3]. В съавторство с проф. Илия Буюклиев в [P1] е описана трансформацията на Уолш-Адамар и методите, предложени от Гуд [42], за разлагане на Кронекерова степен на матрица в произведение от разредени матрици.

Глава 2

Алгоритъм за пресмятане на тегловно разпределение на линеен код над крайно просто поле чрез характеристичен вектор

В тази глава се разглежда крайното просто поле $\mathbb{F}_q = \mathbb{Z}_q = \{0, 1, 2, \ldots, q-1\}$ с фиксирана наредба на елементите $\alpha_0 = 0, \alpha_1 = 1, \alpha_2, \ldots, \alpha_{q-1}$.

С $\boldsymbol{\alpha}$ се означава векторът $(\alpha, \alpha, \ldots, \alpha) = \alpha(1, 1, \ldots, 1)$, състоящ се от едни и същи координати, със съответната подразбираща се дължина.

2.1 Характеристичен вектор на линеен код

Специален тип на пораждащата матрица на симплекс кода $\mathcal{S}_{q,k}$, който ще бъде основно ползван в разработката, се дефинира рекурентно с равенствата:

$$G_1 = (1)\,, \quad G_{k+1} = \begin{pmatrix} \mathbf{0} & \boldsymbol{\alpha_1} & \boldsymbol{\alpha_2} & \ldots & \boldsymbol{\alpha_{q-1}} & 1 \\ G_k & G_k & G_k & & G_k & \mathbf{0}^{\mathrm{T}} \end{pmatrix}\,,\ k \in \mathbb{N}. \tag{2.1}$$

Дефиниция 2.1. *Характеристичен вектор* на линейния $[n, k]_q$ код C по отношение на пораждащата матрица G е векторът

$$\chi(C, G) = (\chi_1, \chi_2, \ldots, \chi_{\theta(q,k)}) \in \mathbb{Z}^{\theta(q,k)}, \tag{2.2}$$

където χ_u е броят на стълбовете на G, които са равни или пропорционални (с ненулев коефициент) на u-тия стълб на матрицата G_k, $u = 1, 2, \ldots, \theta(q, k)$.

По-долу, за краткост характеристичният вектор се означава само с χ, когато кодът C и пораждащата матрица G са ясни от контекста.

Сумата $\sum_{u=1}^{\theta(q,k)} \chi_u$ е броят на ненулевите стълбове на G. При линейни кодове с пълна дължина това е точно дължината на кода.

Даден линеен код може да има различни характеристични вектори в зависимост от избора на пораждаща матрица.

Характеристичният вектор определя клас от мономиално еквивалентни линейни кодове с пълна дължина, които имат едно и също тегловно разпределение.

Нека G е пораждаща матрица на линейния $[n,k]_q$ код C. Пресмятането на характеристичния вектор на кода по отношение на G (съгласно дефиниция 2.1) може да се направи чрез Алгоритъм 1, който връща позицията на стълба от G или негов пропорционален в матрицата G_k. Тъй като алгоритъмът трябва да се приложи за всеки стълб от G, сложността на тази част е $O(nk)$.

Алгоритъм 1 Генериране на характеристичен вектор от пораждаща матрица

Вход: естествени числа q и k, целочислен масив θ, където $\theta[l] = \theta(q,l)$, $l = 1,\ldots,k$, ненулев вектор $g \in \mathbb{F}_q^k$ // *g е стълб от пораждащата матрица G.*

Изход: позицията u на g или пропорционален на g вектор в матрицата G_k

1: $l_1 = 0$
2: $l_2 = k$
3: $u = 0$
4: **while** $l_2 > 0$ **do**
5: **if** $u = 0$ **then**
6: **if** $g[l_2] \neq 0$ **then**
7: $t = g[l_2]^{-1}$ // *в* $\mathbb{F}_q$
8: $u = \theta[l_1 + 1]$
9: **end if**
10: **else**
11: $a = t * g[l_2]$ // *умножение в* $\mathbb{F}_q$
12: $u = Num(a) * \theta[l_1] + u$ // *в* $\mathbb{Z}$, *като* $a = \alpha_{Num(a)}$ *в* $\mathbb{F}_q$
13: **end if**
14: $l_1 = l_1 + 1$
15: $l_2 = l_2 - 1$
16: **end while**

Кодовите думи на даден линеен код се получават от всевъзможните линейни комбинации на редовете на някоя пораждаща матрица G. Всички нену-

леви кодови думи могат лесно да бъдат получени чрез умножение на матрица с вектор, а именно:

$$\begin{pmatrix} G_k^{\mathrm{T}} \\ \alpha_2 G_k^{\mathrm{T}} \\ \vdots \\ \alpha_{q-1} G_k^{\mathrm{T}} \end{pmatrix} \cdot G = \begin{pmatrix} G_k^{\mathrm{T}} \cdot G \\ \alpha_2 G_k^{\mathrm{T}} \cdot G \\ \vdots \\ \alpha_{q-1} G_k^{\mathrm{T}} \cdot G \end{pmatrix}, \tag{2.3}$$

където $\mathbb{F}_q = \{0, 1, \alpha_2, \ldots, \alpha_{q-1}\}$. За намиране на тегловното разпределение на кода C, достатъчно е да се пресметнат теглата на редовете на матрицата $G_k^{\mathrm{T}} \cdot G$.

Нека $M_k = G_k^{\mathrm{T}} \cdot G_k$, $k \in \mathbb{N}$, като умножението е над $\mathbb{F}_q$. По-долу с $\mathcal{N}(M_k)$ е означена матрицата, получена от M_k чрез заместване на ненулевите елементи с 1 (нормализирана матрица).

Лема 2.1. *Нека C е линеен $[n, k]_q$ код с пораждаща матрица G и χ е характеристичният вектор на C по отношение на G. Тогава теглото по Хеминг на i-тия ред на матрицата $G_k^{\mathrm{T}} \cdot G$ (умножението е над $\mathbb{F}_q$) е i-тият елемент на вектора стълб $\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}}$ (умножението е над $\mathbb{Z}$), $i = 1, \ldots, \theta(q, k)$.*

Доказателство. Нека за краткост $\theta = \theta(q, k)$, $g_1, \ldots, g_\theta$ са стълбовете на G_k и $g'_1, \ldots, g'_n$ са стълбовете на G. Тогава g_i^{T} е i-тият ред на G_k^{T}, откъдето $g_i^{\mathrm{T}} \cdot G = (\langle g_i, g'_1 \rangle, \ldots, \langle g_i, g'_n \rangle)$ е i-тият ред на $G_k^{\mathrm{T}} \cdot G$ и $g_i^{\mathrm{T}} \cdot G_k = (\langle g_i, g_1 \rangle, \ldots, \langle g_i \cdot g_\theta \rangle)$ е i-тият ред на M_k. Съгласно дефиниция 2.1 точно χ_u стълба на G са пропорционални на g_u, $u = 1, \ldots, \theta$. Оттук следва, че $\mathrm{wt}(g_i^{\mathrm{T}} \cdot G) = \sum_{u=1}^{\theta} \chi_u \mathcal{N}(\langle g_i, g_u \rangle)$ (събиране над $\mathbb{Z}$), където $\mathcal{N}(\langle g_i, g_u \rangle) = 0$ за $\langle g_i, g_u \rangle = 0$ и $\mathcal{N}(\langle g_i, g_u \rangle) = 1$ в противен случай. От друга страна, $\mathcal{N}(\langle g_i, g_u \rangle)$ е елементът, намиращ се на i-и ред и u-ти стълб в матрицата $\mathcal{N}(M_k)$. Следователно

$$\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}} = \left(\sum_{u=1}^{\theta} \chi_u \mathcal{N}(\langle g_1, g_u \rangle), \ldots, \sum_{u=1}^{\theta} \chi_u \mathcal{N}(\langle g_\theta, g_u \rangle) \right)^{\mathrm{T}}$$

и $\mathrm{wt}(g_i^{\mathrm{T}} \cdot G)$ е i-тият елемент на този вектор. □

Дефиниция 2.2. Подмножество U на линейния $[n, k]_q$ код C се нарича *максимално подмножество от непропорционални кодови думи* на C, ако притежава следните свойства:

1. всеки две кодови думи $c, c' \in U$ не са пропорционални помежду си, т. е. не съществуват скалари $\lambda, \lambda' \in \mathbb{F}_q$, така че $c' = \lambda c$ и $c = \lambda' c'$;

2. за всяка кодова дума $c' \in C \backslash \{\mathbf{0}\}$, съществува кодова дума $c \in U$, такава че c' е пропорционална на c, т. е. съществува скалар $\lambda \in \mathbb{F}_q$, за който $c' = \lambda c$.

Лема 2.1 и (2.3) показват, че координатите на вектора $\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}}$ са теглата на всички кодови думи от максимално подмножество U от непропорционални кодови думи. Следователно, при $w \neq 0$, ако N_w е мощността на множеството $\{c \in U \mid \mathrm{wt}(c) = w\}$, то броят на кодовите думи с тегло w в кода е $A_w = (q-1)N_w$. Така, с едно умножение на матрица по вектор може да се получи тегловното разпределение на даден код, без да се изчисляват всички кодови думи.

От (2.1) се получава рекурентна връзка за матриците M_k, а именно: $M_1 = (1)$ и за всяко $k \in \mathbb{Z}, k \geq 2$

$$M_k = \begin{pmatrix} M_{k-1} & M_{k-1} & \dots & M_{k-1} & \mathbf{0}^{\mathrm{T}} \\ M_{k-1} & M_{k-1} + J & \dots & M_{k-1} + \alpha_{q-1}J & \mathbf{1}^{\mathrm{T}} \\ M_{k-1} & M_{k-1} + \alpha_2 J & \dots & M_{k-1} + \alpha_2\alpha_{q-1}J & \boldsymbol{\alpha_2}^{\mathrm{T}} \\ \vdots & & & & \\ M_{k-1} & M_{k-1} + \alpha_{q-1}J & \dots & M_{k-1} + \alpha_{q-1}^2 J & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{1} & \dots & \boldsymbol{\alpha_{q-1}} & 1 \end{pmatrix}. \tag{2.4}$$

Матрицата J в горната формула е $\theta(q, k-1) \times \theta(q, k-1)$ матрица, състояща се от единици.

Структурата на матриците G_k е специално избрана, за да има само събиране на матрици в рекурентната връзка (2.4).

За съжаление, няма удобна рекурентна връзка за матриците $\mathcal{N}(M_k)$. За да се избегне този проблем, в следващия раздел е въведено понятието характеристично разпределение.

2.2 Характеристично разпределение

Дефиниция 2.3. Нека $\chi = (\chi_1, \dots, \chi_t) \in \mathbb{Z}^t$ и $b = (b_1, \dots, b_t) \in \mathbb{F}_q^t$, $t \in \mathbb{N}$. *Характеристично разпределение* на вектора b по отношение на χ е векторът

$$b^{[\chi]} = (\mu_0, \mu_1, \dots, \mu_{q-1}) \in \mathbb{Z}^q,$$

където μ_j е сумата от координатите χ_u на вектора χ, такива че $b_u = \alpha_j$, $1 \leq u \leq t$, $j = 0, 1, \dots, q-1$. Ако няма координати на b, които са равни на α_j, то $\mu_j = 0$.

С други думи, $b^{[\chi]}$ може да бъде изчислен от вектора

$$b' = (\underbrace{b_1, \dots, b_1}_{\chi_1}, \dots, \underbrace{b_t, \dots, b_t}_{\chi_t}),$$

където има μ_j появявания на α_j във вектора b'. Според [57, p. 142] с $b^{[\chi]} = \mathrm{comp}(b')$ е означена композицията на b'.

Пример 2.1. Ако $q = 3$, $\chi = (3, 4, 5, 1, 3)$ и $b = (1, 2, 2, 0, 0)$, то $b^{[\chi]} = (1 + 3, 3, 4 + 5) = (4, 3, 9)$.

Пример 2.2. Ако $q = 5$, $\chi = (8, 5, 3, 1)$ и $b = (3, 0, 1, 1)$, то $b^{[\chi]} = (5, 4, 0, 8, 0)$.

Понятието характеристично разпределение може да се представи с помощта на заместванията. Нека $b_{0\uparrow}$ означава векторът, получен от b чрез заместване на нулевите му елементи с 1 и всички останали елементи с 0. Аналогично, $b_{\alpha\uparrow}$ означава векторът, получен от b чрез заместване на всички координати, равни на α, с 1 и всички останали координати с 0. Характеристичното разпределение се състои от скаларните произведения на $b_{\alpha\uparrow}$ и χ^{T} над $\mathbb{Z}$. Нека с $b_\uparrow$ е означена матрицата, чиито редове са $b_{\alpha_j\uparrow}$, $j = 0, \ldots, q-1$. Тогава

$$b^{[\chi]} = \left(b_{0\uparrow} \cdot \chi^{\mathrm{T}}, b_{1\uparrow} \cdot \chi^{\mathrm{T}}, \ldots, b_{\alpha_{q-1}\uparrow} \cdot \chi^{\mathrm{T}}\right) = \left(\begin{pmatrix} b_{0\uparrow} \\ b_{1\uparrow} \\ \ldots \\ b_{\alpha_{q-1}\uparrow} \end{pmatrix} \cdot \chi^{\mathrm{T}}\right)^{\mathrm{T}} = (b_\uparrow \cdot \chi^{\mathrm{T}})^{\mathrm{T}} = \chi \cdot b_\uparrow^{\mathrm{T}}.$$

Пример 2.3. Нека $q = 3$, $\chi = (6, 4, 2, 10)$ и $b = (0, 1, 1, 2)$. Тогава

$$\begin{aligned} b_{0\uparrow} &= (1, 0, 0, 0) \\ b_{1\uparrow} &= (0, 1, 1, 0) \\ b_{2\uparrow} &= (0, 0, 0, 1) \end{aligned}$$

$$\Rightarrow b^{[\chi]} = \left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 4 \\ 2 \\ 10 \end{pmatrix}\right)^{\mathrm{T}} = (6, 6, 10).$$

В следващото твърдение са изброени някои елементарни свойства на характеристичното разпределение.

Твърдение 2.2. *Нека $\chi = (\chi_1, \ldots, \chi_t) \in \mathbb{Z}^t$ и $b = (b_1, \ldots, b_t) \in \mathbb{F}_q^t$, $t \in \mathbb{N}$. Тогава характеристичното разпределение $b^{[\chi]}$ на вектора b по отношение на χ има следните свойства:*

1. $\sum_{j=0}^{q-1} \mu_j = \sum_{u=1}^{t} \chi_u$. *Това свойство обяснява, че характеристичното разпределение е специфично разпределение на координатите на χ.*

2. *Ако всички координати на b са равни на $\alpha_j \in \mathbb{F}_q$, $j = 0, \ldots, q-1$, то характеристичното разпределение $b^{[\chi]}$ се състои от нули с изключение на $(j+1)$-вия елемент, който е равен на сумата от координатите на вектора χ,*
$\mathbf{1}^{[\chi]} = \big(0, \sum_{u=1}^{t} \chi_u, 0, \ldots, 0\big)$, $\mathbf{0}^{[\chi]} = \big(\sum_{u=1}^{t} \chi_u, 0, \ldots, 0\big)$.

3. *Ако $\mathcal{N}(b)$ е получен от b чрез заместване на всички ненулеви координати с 1, то*

$$\mathcal{N}(b)^{[\chi]} = \left(\mu_0, \sum_{j=1}^{q-1} \mu_j, 0, \ldots, 0\right), \quad \mathcal{N}(b) \cdot \chi^{\mathrm{T}} = \sum_{j=1}^{q-1} \mu_j = \left(\sum_{u=1}^{t} \chi_u\right) - \mu_0$$

(над $\mathbb{Z}$).

4. *При добавяне на 1 към всяка координата в b (над $\mathbb{F}_q$) характеристичното разпределение се променя, както при операцията циклично преместване надясно (означава се с SR), т. е.*

$$(b + \mathbf{1})^{[\chi]} = SR(b^{[\chi]}).$$

5. *Ако се добави елементът $\alpha_j = \underbrace{1 + 1 + \cdots + 1}_{j} \in \mathbb{F}_q$ към всяка координата на b, новото характеристично разпределение може да се получи от старото чрез прилагане на операцията SR j пъти, т. е.*

$$(b + \boldsymbol{\alpha_j})^{[\chi]} = \underbrace{SR(\ldots SR}_{j}(b^{[\chi]})) = SR_j(b^{[\chi]}).$$

6. *Ако $s, t \in \mathbb{N}$, $\chi' \in \mathbb{Z}^s$, $\chi'' \in \mathbb{Z}^t$, $b' \in \mathbb{F}_q^s$, $b'' \in \mathbb{F}_q^t$, $\chi = (\chi'|\chi'')$, $b = (b'|b'')$, то*

$$b^{[\chi]} = b'^{[\chi']} + b''^{[\chi'']}.$$

Горните свойства следват директно от дефиницията за характеристично разпределение.

Пример 2.4. Нека $q = 5$, $\chi = (9, 1, 4, 2, 6)$ и $b = (1, 0, 3, 2, 1)$. Тогава $b^{[\chi]} = (1, 15, 2, 4, 0)$,

$$\mathcal{N}(b) = (1, 0, 1, 1, 1), \quad \mathcal{N}(b)^{[\chi]} = (1, 21, 0, 0, 0), \quad \mathcal{N}(b) \cdot \chi^{\mathrm{T}} = 21.$$

В допълнение

$$b+\mathbf{1}=(2,1,4,3,2), \quad (b+\mathbf{1})^{[\chi]}=(0,1,15,2,4),$$
$$b+\mathbf{2}=(3,2,0,4,3), \quad (b+\mathbf{2})^{[\chi]}=(4,0,1,15,2),$$
$$b+\mathbf{3}=(4,3,1,0,4), \quad (b+\mathbf{3})^{[\chi]}=(2,4,0,1,15).$$

Пример 2.5. Нека $q=3$,

$$\chi=(\underbrace{3,4,5,1,3}_{\chi'},\underbrace{6,4,2,10}_{\chi''}), \quad b=(\underbrace{1,2,2,0,0}_{b'},\underbrace{0,1,1,2}_{b''}),$$

$$b'^{[\chi']}=(4,3,9), \quad b''^{[\chi'']}=(6,6,10) \quad \Longrightarrow b^{[\chi]}=(10,9,19)$$

Следствието по-долу описва горните свойства чрез матричното представяне.

Следствие 2.3. *Нека $\chi=(\chi_1,\ldots,\chi_t)\in\mathbb{Z}^t$ и $b=(b_1,\ldots,b_t)\in\mathbb{F}_q^t$, $t\in\mathbb{N}$. Тогава*

1. *Всеки стълб на матрицата $b_\uparrow$ съдържа точно една единица и $q-1$ нули.*
2. *Ако всички координати на b са равни на един и същ елемент α_j, то $(j+1)$-вият ред на матрицата $b_\uparrow$ се състои само от единици, а всички други редове се състоят само от нули.*
3. $b=(0,1,\alpha_2,\ldots,\alpha_{q-1})\cdot b_\uparrow$, $\mathcal{N}(b)=(0,1,1,\ldots,1)\cdot b_\uparrow$.
4. *Съществува пермутационна матрица P_1, такава че $(b+\mathbf{1})^{[\chi]}=b^{[\chi]}\cdot P_1^{\mathrm{T}}$. Казва се, че матрицата P_1 реализира операцията SR.*
5. *Съществува пермутационна матрица $P_{\alpha_j}=P_1^j$, такава че $(b+\boldsymbol{\alpha_j})^{[\chi]}=b^{[\chi]}\cdot P_{\alpha_j}^{\mathrm{T}}$, $\alpha_j\in\mathbb{F}_q$, $j\geq 1$. Казва се, че матрицата P_{α_j} реализира операцията SR_{α_j}.*
6. $b^{[\chi]}=b'^{[\chi']}+b''^{[\chi'']}=(b'^{[\chi']}|b''^{[\chi'']})\cdot\begin{pmatrix} I_q \\ I_q \end{pmatrix}$, *където I_q е единичната матрица от ред q.*

Следва дефиниция на понятието характеристично разпределение на матрица.

Дефиниция 2.4. Нека $s,t\in\mathbb{N}$, $\chi\in\mathbb{Z}^t$, $B\in\mathbb{F}_q^{s\times t}$ и $B_1,\ldots,B_s$ са редовете на матрицата B. *Характеристично разпределение* на матрицата B по отношение на вектора χ е матрицата $B^{[\chi]}\in\mathbb{Z}^{s\times q}$, чиито редове са $B_1^{[\chi]},\ldots,B_s^{[\chi]}$.

В допълнение, ако B' и B'' са матрици с по t стълба, то

$$\begin{pmatrix} B' \\ B'' \end{pmatrix}^{[\chi]} = \begin{pmatrix} B'^{[\chi]} \\ B''^{[\chi]} \end{pmatrix}$$

Пример 2.6. Нека $q = 3$ и $\chi = (1, 4, 3, 2)$. Тогава

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 2 \end{pmatrix} \quad \Rightarrow \quad B^{[\chi]} = \begin{pmatrix} 2 & 8 & 0 \\ 3 & 3 & 4 \\ 4 & 1 & 5 \end{pmatrix}.$$

Твърдение 2.2 може естествено да се обобщи за характеристичното разпределение на матрици.

По-нататък в текста, с χ ще се означава характеристичният вектор на линеен $[n, k]_q$ код с пълна дължина C по отношение на негова пораждаща матрица G. Според следващата теорема изчисляването на тегловното разпределение на код се свежда до пресмятането на $\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}}$.

Теорема 2.4. *Нека C е линеен $[n, k]_q$ код с пълна дължина и χ е характеристичен вектор на кода C по отношение на някоя негова пораждаща матрица. Поредната i-та координата на $\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}}$ е равна на $n - \mu_0$, където μ_0 е първата координата на характеристичното разпределение $c_i^{[\chi]} = (\mu_0, \mu_1, \ldots, \mu_{q-1})$ на i-тия ред c_i на матрицата M_k по отношение на χ.*

Доказателство. Според трето свойство от твърдение 2.2,

$$\mathcal{N}(c_i) \cdot \chi^{\mathrm{T}} = \sum_{j=1}^{q-1} \mu_j = \left(\sum_{u=1}^{\theta(q,k)} \chi_u \right) - \mu_0.$$

Тъй като C е линеен код с пълна дължина, то $\sum_{u=1}^{\theta(q,k)} \chi_u = n$ и следователно $\mathcal{N}(c_i) \cdot \chi^{\mathrm{T}} = n - \mu_0$. □

Нещо повече. Матрицата $M_k^{[\chi]}$ дава достатъчно информация за пълната тегловна функция на мономиално еквивалентен на разглеждания линеен код. Както по-рано беше отбелязано, характеристичният вектор определя клас от мономиално еквивалентни линейни $[n, k]_q$ кодове с пълна дължина. За представител C'' на такъв клас може да се избере линеен код с пораждаща матрица, чиито стълбове са измежду стълбовете на матрицата G_k (евентуално повтарящи се). Всеки ред на $M_k^{[\chi]}$ съответства на някоя кодова дума c от максимално подмножество U от непропорционални кодови думи на C''. Елементите на реда показват броя на координатите на c, равни съответно на α_j, за всяко $j = 0, 1, \ldots, q-1$.

Пример 2.7. Нека C е троичен линеен код с пълна дължина и характеристичен вектор

$$\chi = (1, 2, 0, 4, 3, 2, 2, 1, 0, 0, 1, 1, 3),$$

откъдето е ясно, че дължината на кода е $n = 20$, а размерността е $k = 3$. Тогава $\mathcal{N}(M_k) \cdot \chi^{\mathrm{T}} = (11, 14, 13, 13, 15, 17, 15, 16, 9, 16, 13, 15, 13)$. Следователно тегловната функция на C е $W(z) = 1 + 2(z^9 + z^{11} + 4z^{13} + z^{14} + 3z^{15} + 2z^{16} + z^{17})$. Освен това,

$$M_3^{[\chi]} = \begin{pmatrix} 10 & 11 & 0 \\ 6 & 10 & 4 \\ 7 & 4 & 9 \\ 7 & 10 & 3 \\ 5 & 7 & 8 \\ 3 & 10 & 7 \\ 5 & 8 & 7 \\ 4 & 13 & 3 \\ 11 & 4 & 5 \\ 4 & 8 & 8 \\ 7 & 4 & 9 \\ 5 & 8 & 7 \\ 11 & 8 & 1 \end{pmatrix}$$

и ако C'' е представител на съответния клас мономиално еквивалентни линейни кодове с пълна дължина, определен от χ, то

$$W_{C''}(z_0, z_1, z_2) = 1 + z_0^{10}(z_1^{11} + z_2^{11}) + z_0^6(z_1^{10}z_2^4 + z_1^4 z_2^{10}) + \cdots + z_0^{11}(z_1^8 z_2 + z_1 z_2^8).$$

Нека характеристичният вектор χ на линейния $[n, k]_q$ код C е разделен на $q + 1$ части, както следва

$$\chi = \left(\chi^{(0)} | \chi^{(1)} | \ldots | \chi^{(q-1)} | \chi^{(q)}\right), \tag{2.5}$$

където $\chi^{(j)} \in \mathbb{Z}^{\theta(q,k-1)}$, $j = 0, \ldots, q-1$ и $\chi^{(q)} \in \mathbb{Z}$. Известно е, че $\theta(q, k) = q\theta(q, k-1) + 1$. Тогава е в сила следната рекурентна връзка:

$M_k^{[\chi]} =$

$$\begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} + & M_{k-1}^{[\chi^{(1)}]} & + \cdots + & M_{k-1}^{[\chi^{(q-1)}]} & + & \mathbf{0}^{\mathrm{T}[\chi^{(q)}]} \\ M_{k-1}^{[\chi^{(0)}]} + & (M_{k-1} + J)^{[\chi^{(1)}]} & + \cdots + & (M_{k-1} + \alpha_{q-1}J)^{[\chi^{(q-1)}]} & + & \mathbf{1}^{\mathrm{T}[\chi^{(q)}]} \\ M_{k-1}^{[\chi^{(0)}]} + & (M_{k-1} + \alpha_2 J)^{[\chi^{(1)}]} & + \cdots + & (M_{k-1} + \alpha_2\alpha_{q-1}J)^{[\chi^{(q-1)}]} & + & \boldsymbol{\alpha_2}^{\mathrm{T}[\chi^{(q)}]} \\ & & \vdots & & & \\ M_{k-1}^{[\chi^{(0)}]} + & (M_{k-1} + \alpha_{q-1}J)^{[\chi^{(1)}]} & + \cdots + & (M_{k-1} + \alpha_{q-1}^2 J)^{[\chi^{(q-1)}]} & + & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}[\chi^{(q)}]} \\ \mathbf{0}^{[\chi^{(0)}]} + & \mathbf{1}^{[\chi^{(1)}]} & + \cdots + & \boldsymbol{\alpha_{q-1}}^{[\chi^{(q-1)}]} & + & 1^{[\chi^{(q)}]} \end{pmatrix}$$

$$\Rightarrow M_k^{[\chi]} =$$

$$\begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} & + & M_{k-1}^{[\chi^{(1)}]} & +\cdots+ & M_{k-1}^{[\chi^{(q-1)}]} & + & \mathbf{0}^{\mathrm{T}[\chi^{(q)}]} \\ M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_{q-1}}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \mathbf{1}^{\mathrm{T}[\chi^{(q)}]} \\ M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}_{\alpha_2}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_2\alpha_{q-1}}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \boldsymbol{\alpha_2}^{\mathrm{T}[\chi^{(q)}]} \\ & & & \vdots & & & \\ M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}_{\alpha_{q-1}}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_{q-1}^2}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}[\chi^{(q)}]} \\ \mathbf{0}^{[\chi^{(0)}]} & + & \mathbf{1}^{[\chi^{(1)}]} & +\cdots+ & \boldsymbol{\alpha_{q-1}}^{[\chi^{(q-1)}]} & + & 1^{[\chi^{(q)}]} \end{pmatrix} \tag{2.6}$$

Така може да се използват само пермутации и събиране, за да се изчисли $M_k^{[\chi]}$ от $M_{k-1}^{[\chi^{(0)}]}, M_{k-1}^{[\chi^{(1)}]}, \ldots, M_{k-1}^{[\chi^{(q-1)}]}$ и $\chi^{(q)}$. Освен това, $\mathbf{1}^{[\chi]}, \ldots, \boldsymbol{\alpha_{q-1}}^{[\chi]}$ могат да се получат от $\mathbf{0}^{[\chi]}$ чрез операцията SR.

Пример 2.8. Ако $q = 3$, рекурентната връзка (2.6) е

$$M_k^{[\chi]} = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} & + & M_{k-1}^{[\chi^{(1)}]} & + & M_{k-1}^{[\chi^{(2)}]} & + & \mathbf{0}^{\mathrm{T}[\chi^{(3)}]} \\ M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}(M_{k-1}^{[\chi^{(1)}]}) & + & \mathrm{SR}_2(M_{k-1}^{[\chi^{(2)}]}) & + & \mathbf{1}^{\mathrm{T}[\chi^{(3)}]} \\ M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}_2(M_{k-1}^{[\chi^{(1)}]}) & + & \mathrm{SR}(M_{k-1}^{[\chi^{(2)}]}) & + & \mathbf{2}^{\mathrm{T}[\chi^{(3)}]} \\ \mathbf{0}^{[\chi^{(0)}]} & + & \mathbf{1}^{[\chi^{(1)}]} & + & \mathbf{2}^{[\chi^{(2)}]} & + & 1^{[\chi^{(3)}]} \end{pmatrix} \tag{2.7}$$

Нека $k = 3$ и $\chi = (0, 4, 3, 2, 0, 8, 5, 1, 1, 4, 3, 2, 3)$. Тогава $\theta(3, 3) = 13$. Съгласно (2.5) векторът χ се разделя на 4 части

$$\chi^{(0)} = (0, 4, 3, 2), \quad \chi^{(1)} = (0, 8, 5, 1), \quad \chi^{(2)} = (1, 4, 3, 2), \quad \chi^{(3)} = 3.$$

За да се пресметне $M_3^{[\chi]}$, трябва да се изчислят $M_2^{[\chi^{(0)}]}$, $M_2^{[\chi^{(1)}]}$ и $M_2^{[\chi^{(2)}]}$, където

$$M_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 1 \end{pmatrix}.$$

От дефиниции 2.3 и 2.4 се получава

$$M_2^{[\chi^{(0)}]} = \begin{pmatrix} 2 & 7 & 0 \\ 3 & 2 & 4 \\ 4 & 0 & 5 \\ 0 & 6 & 3 \end{pmatrix}, \quad M_2^{[\chi^{(1)}]} = \begin{pmatrix} 1 & 13 & 0 \\ 5 & 1 & 8 \\ 8 & 0 & 6 \\ 0 & 9 & 5 \end{pmatrix}, \quad M_2^{[\chi^{(2)}]} = \begin{pmatrix} 2 & 8 & 0 \\ 3 & 3 & 4 \\ 4 & 1 & 5 \\ 1 & 6 & 3 \end{pmatrix}.$$

Следва пресмятането на $M_3^{[\chi]}$ от $M_2^{[\chi^{(0)}]}$, $M_2^{[\chi^{(1)}]}$, $M_2^{[\chi^{(2)}]}$ и $\chi^{(3)}$ по (2.7):

$$
\begin{aligned}
M_3^{[\chi]} &= \left(\begin{pmatrix} 2 & 7 & 0 \\ 3 & 2 & 4 \\ 4 & 0 & 5 \\ 0 & 6 & 3 \end{pmatrix} + \begin{pmatrix} 1 & 13 & 0 \\ 5 & 1 & 8 \\ 8 & 0 & 6 \\ 0 & 9 & 5 \end{pmatrix} + \begin{pmatrix} 2 & 8 & 0 \\ 3 & 3 & 4 \\ 4 & 1 & 5 \\ 1 & 6 & 3 \end{pmatrix} + \begin{pmatrix} 3 & 0 & 0 \\ 3 & 0 & 0 \\ 3 & 0 & 0 \\ 3 & 0 & 0 \end{pmatrix} \right. \\
&\quad \begin{pmatrix} 2 & 7 & 0 \\ 3 & 2 & 4 \\ 4 & 0 & 5 \\ 0 & 6 & 3 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 13 \\ 8 & 5 & 1 \\ 6 & 8 & 0 \\ 5 & 0 & 9 \end{pmatrix} + \begin{pmatrix} 8 & 0 & 2 \\ 3 & 4 & 3 \\ 1 & 5 & 4 \\ 6 & 3 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 3 & 0 \\ 0 & 3 & 0 \\ 0 & 3 & 0 \\ 0 & 3 & 0 \end{pmatrix} \\
&\quad \begin{pmatrix} 2 & 7 & 0 \\ 3 & 2 & 4 \\ 4 & 0 & 5 \\ 0 & 6 & 3 \end{pmatrix} + \begin{pmatrix} 13 & 0 & 1 \\ 1 & 8 & 5 \\ 0 & 6 & 8 \\ 9 & 5 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 8 \\ 4 & 3 & 3 \\ 5 & 4 & 1 \\ 3 & 1 & 6 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 3 \\ 0 & 0 & 3 \\ 0 & 0 & 3 \\ 0 & 0 & 3 \end{pmatrix} \\
&\quad \left. \begin{pmatrix} 9 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 14 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 10 \end{pmatrix} + \begin{pmatrix} 0 & 3 & 0 \end{pmatrix} \right) \\
&= \left(\begin{array}{rrr} 8 & 28 & 0 \\ 14 & 6 & 16 \\ 19 & 1 & 16 \\ 4 & 21 & 11 \\ \hline 10 & 11 & 15 \\ 14 & 14 & 8 \\ 11 & 16 & 9 \\ 11 & 12 & 13 \\ \hline 15 & 9 & 12 \\ 8 & 13 & 15 \\ 9 & 10 & 17 \\ 12 & 12 & 12 \\ \hline 9 & 17 & 10 \end{array} \right).
\end{aligned}
$$

По-долу се въвежда друго важно понятие за характеристичното разпределение на матрицата M_k, което се ползва в разработените алгоритми.

Дефиниция 2.5. Нека $k \in \mathbb{N}$ и $\chi = (\chi_1, \ldots, \chi_{\theta(q,k)}) \in \mathbb{Z}^{\theta(q,k)}$. *Частично характеристично разпределение* $M_k^{[\chi]}(l)$ за $l = 1, \ldots, k$ се дефинира рекурсивно, както следва

1. $M_k^{[\chi]}(k) = M_k^{[\chi]}$.
2. Ако $1 \leq l < k$ и векторът χ се разделя на $q+1$ части, както в (2.5), то

$M_k^{[\chi]}(l)$ се пресмята по формулата

$$M_k^{[\chi]}(l) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]}(l) \\ M_{k-1}^{[\chi^{(1)}]}(l) \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]}(l) \\ M_1^{[\chi^{(q)}]} \end{pmatrix}.$$

Матрицата $M_k^{[\chi]}(1)$ е с размерност $\theta(q,k) \times q$ и редове $M_1^{[\chi_u]}$, за $u = 1, \dots, \theta(q,k)$. Тъй като $M_1 = (1)$, стълбовете на матрицата $M_k^{[\chi]}(1)$ са нулеви вектори с изключение на втория, който е равен на χ^{T}.

Последният ред на матрицата $M_k^{[\chi]}(l)$ за $l = 1, \dots, k-1$ е един и същ, именно $M_1^{[\chi^{(q)}]}$. Освен това, редът преди последния в $M_k^{[\chi]}(l)$ е един и същ за $l = 1, \dots, k-2$. Действително, за всяко $l < k$ има някои редове, които се запазват във всички матрици $M_k^{[\chi]}(l')$ за $1 \le l' < l$. Тези редове се наричат *неактивни редове*. Има точно $\theta(q, k-l)$ неактивни реда в $M_k^{[\chi]}(l)$, $l = 2, \dots, k-1$.

Пример 2.9. Нека $q = 3$, $k = 3$ и $\chi = (0,4,3,2,0,8,5,1,1,4,3,2,3)$. Тогава

$$M_3^{[\chi]}(1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 3 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 3 & 0 \\ 0 & 2 & 0 \\ 0 & 3 & 0 \end{pmatrix}, \; M_3^{[\chi]}(2) = \left(\begin{array}{ccc} 2 & 7 & 0 \\ 3 & 2 & 4 \\ 4 & 0 & 5 \\ 0 & 6 & 3 \\ \hline 1 & 13 & 0 \\ 5 & 1 & 8 \\ 8 & 0 & 6 \\ 0 & 9 & 5 \\ \hline 2 & 8 & 0 \\ 3 & 3 & 4 \\ 4 & 1 & 5 \\ 1 & 6 & 3 \\ \hline 0 & 3 & 0 \end{array}\right), \; M_3^{[\chi]}(3) = \begin{pmatrix} 8 & 28 & 0 \\ 14 & 6 & 16 \\ 19 & 1 & 16 \\ 4 & 21 & 11 \\ 10 & 11 & 15 \\ 14 & 14 & 8 \\ 11 & 16 & 9 \\ 11 & 12 & 13 \\ 15 & 9 & 12 \\ 8 & 13 & 15 \\ 9 & 10 & 17 \\ 12 & 12 & 12 \\ 9 & 17 & 10 \end{pmatrix}.$$

Пример 2.10. Нека $q = 3$, $k = 4$. Характеристичният вектор χ е разделен на части според (2.5):

$$\chi = (\underbrace{\chi^{(0,0)}|\chi^{(0,1)}|\chi^{(0,2)}|\chi^{(0,3)}}_{\chi^{(0)}}|\underbrace{\chi^{(1,0)}|\chi^{(1,1)}|\chi^{(1,2)}|\chi^{(1,3)}}_{\chi^{(1)}}|\underbrace{\chi^{(2,0)}|\chi^{(2,1)}|\chi^{(2,2)}|\chi^{(2,3)}}_{\chi^{(2)}}|\chi^{(3)}).$$

В $M_4^{[\chi]}(3)$ има един неактивен ред, в $M_4^{[\chi]}(2)$ има 4 неактивни реда:

$$M_4^{[\chi]}(3) = \begin{pmatrix} M_3^{[\chi^{(0)}]}(3) \\ M_3^{[\chi^{(1)}]}(3) \\ M_3^{[\chi^{(2)}]}(3) \\ M_1^{[\chi^{(3)}]} \end{pmatrix}, \quad M_4^{[\chi]}(2) = \begin{pmatrix} M_3^{[\chi^{(0)}]}(2) \\ M_3^{[\chi^{(1)}]}(2) \\ M_3^{[\chi^{(2)}]}(2) \\ M_1^{[\chi^{(3)}]} \end{pmatrix} = \begin{pmatrix} M_2^{[\chi^{(0,0)}]}(2) \\ M_2^{[\chi^{(0,1)}]}(2) \\ M_2^{[\chi^{(0,2)}]}(2) \\ M_1^{[\chi^{(0,3)}]} \\ M_2^{[\chi^{(1,0)}]}(2) \\ M_2^{[\chi^{(1,1)}]}(2) \\ M_2^{[\chi^{(1,2)}]}(2) \\ M_1^{[\chi^{(1,3)}]} \\ M_2^{[\chi^{(2,0)}]}(2) \\ M_2^{[\chi^{(2,1)}]}(2) \\ M_2^{[\chi^{(2,2)}]}(2) \\ M_1^{[\chi^{(2,3)}]} \\ M_1^{[\chi^{(3)}]} \end{pmatrix}.$$

2.3 Алгоритъм за намиране на характеристично разпределение

В този раздел е представен разработеният алгоритъм за пресмятане на $M_k^{[\chi]}$ с последователно изчисляване на $M_k^{[\chi]}(1)$, $M_k^{[\chi]}(2)$,..., $M_k^{[\chi]}(k-1)$, $M_k^{[\chi]}(k)$. Псевдокод на основната процедура е даден в Алгоритъм 2.

Алгоритъм 3 показва как да се получи $M_k^{[\chi]}(l)$ от $M_k^{[\chi]}(l-1)$. Алгоритъмът се състои от три основни преобразувания, които се наричат Add0, LastRow и AllRows. По-долу тези преобразувания са обяснени при $l = k$. Пресмятането започва с

$$M_k^{[\chi]}(k-1) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} \\ M_{k-1}^{[\chi^{(1)}]} \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]} \\ M_1^{[\chi^{(q)}]} \end{pmatrix} = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} \\ M_{k-1}^{[\chi^{(1)}]} \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]} \\ 0, \chi^{(q)}, 0, \dots, 0 \end{pmatrix}.$$

1. Add0: Първо, прилага се операцията циклично преместване наляво върху последния ред на матрицата $M_k^{[\chi]}(k-1)$. Полученият вектор $\mathrm{lcs}(M_1^{[\chi^{(q)}]}) = (\chi^{(q)}, 0, \dots, 0) = 0^{[\chi^{(q)}]}$ се добавя към всеки ред на матрицата $M_{k-1}^{[\chi^{(1)}]}$.

Алгоритъм 2 Основна процедура

Вход: целите числа q и k и вектор χ с дължина $\theta = \frac{q^k-1}{q-1}$ с целочислени координати

Изход: масив D // $D = M_k^{[\chi]}$

1: $D := M_k^{[\chi]}(1)$; $\theta_1 := 1$
2: **for** $l = 2$ **to** k **do**
3: Инициализира се масив a с дължина k, $a := \mathbf{0}$ // *помощен масив за проследяване на неактивните редове*
4: $\theta_0 := \theta_1$; $\theta_1 := \theta(q,l) = \frac{q^l-1}{q-1} = q * \theta_0 + 1$; $i_1 := 0$
5: **while** $i_1 < \theta$ **do**
6: $i_0 := i_1$ // *индекс на реда преди първия ред от текущата подматрица*
7: $i_1 := i_1 + \theta_1$ // *индекс на последния ред от текущата подматрица*
8: $\mathrm{NewD}(D, i_0, i_1, \theta_0)$ // *Функция за пресмятане на характеристичното разпределение на матрицата M_l по отношение на текущата част от χ*
9: $s := l$; $a[s] := a[s] + 1$
10: **while** $a[s] = q$ **do**
11: $i_1 := i_1 + 1$ // *пропускане на неактивен ред*
12: $a[s] := 0$; $s := s + 1$; $a[s] := a[s] + 1$
13: **end while**
14: **end while**
15: **end for**

$$M_k^{[\chi]}(k-1) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} \\ M_{k-1}^{[\chi^{(1)}]} \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]} \\ 0, \chi^{(q)}, 0, \dots, 0 \end{pmatrix} \longrightarrow \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} \\ M_{k-1}^{[\chi^{(1)}]} + \mathbf{0}^{\mathrm{T}[\chi^{(q)}]} \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]} \\ 0, \chi^{(q)}, 0, \dots, 0 \end{pmatrix}$$

2. LastRow: Пресмята се последният ред на матрицата $M_k^{[\chi]}(k)$, който е равен на

$$\begin{aligned} M_k^{[\chi]}last &= \left(\mathbf{0}^{[\chi^{(0)}]} + \mathbf{1}^{[\chi^{(1)}]} + \dots + \boldsymbol{\alpha_{q-1}}^{[\chi^{(q-1)}]} + 1^{[\chi^{(q)}]}\right) \\ &= \left(\sum_{u=1}^{\theta_0} \chi_u,\ \chi^{(q)} + \sum_{u=\theta_0+1}^{2\theta_0} \chi_u, \dots, \sum_{u=\theta_1-\theta_0}^{\theta_1-1} \chi_u\right), \end{aligned}$$

Алгоритъм 3 Функция NewD(D, i_0, i_1, θ_0)

Вход: Матрица D и цели числа i_0, i_1, θ_0 // *параметри, които фиксират текущата подматрица*

Изход: актуализирана матрицата D // *в частта на текущата подматрица*

1: Инициализиране на помощния масив $TEMP$ с размери $q \times q$
2: **for** $i = 1$ **to** θ_0 **do**
3: $D[i_0 + \theta_0 + i] := D[i_0 + \theta_0 + i] + \mathrm{lcs}(D[i_1])$ // *Преобразуването* Add0
4: **end for**
5: $D[i_1] = (\sum_{i=0}^{q-1} D[i_0{+}1, i], \sum_{i=0}^{q-1} D[i_0{+}\theta_0{+}1, i], \ldots, \sum_{i=0}^{q-1} D[i_0{+}(q{-}1){*}\theta_0{+}1, i])$; // LastRow
6: **for** $i = 1$ **to** θ_0 **do**
7: **for** $j = 0$ **to** $q - 1$ **do**
8: $TEMP[j] := D[i_0 + j * \theta_0 + i]$
9: **end for**
10: $D[i_0 + i] := TEMP[0] + TEMP[1] + \cdots + TEMP[q-1]$
11: **for** $j = 1$ **to** $q - 1$ **do**
12: $D[i_0{+}j{*}\theta_0{+}i] := TEMP[0]{+}SR_{\alpha_j}(TEMP[1]){+}\cdots{+}SR_{\alpha_j\alpha_{q-1}}(TEMP[q{-}1])$ // AllRows
13: **end for**
14: **end for**

където $\theta_0 = \theta(q, k-1)$ и $\theta_1 = \theta(q, k)$. Съществено се използва, че $\theta_1 = q.\theta_0 + 1$. Първото свойство от твърдение 2.2 показва, че $\sum_{u=j\theta_0+1}^{(j+1)\theta_0} \chi_u$ е равно на сумата от координатите на първия ред (и на всеки от следващите $\theta_0 - 1$ реда) на матрицата $M_{k-1}^{[\chi^{(j)}]}$, $j = 0, 1, \ldots, q-1$. Това е основанието, в LastRow да се сумират координатите на първия ред на матриците $M_{k-1}^{[\chi^{(j)}]}$, както и да се поставят тези суми като координати в последния ред на обновената матрица:

$$\begin{aligned} M_k^{[\chi]}last &= \left(\sum_{u=1}^{\theta_0}\chi_u,\ \chi^{(q)} + \sum_{u=\theta_0+1}^{2\theta_0}\chi_u, \ldots, \sum_{u=\theta_1-\theta_0}^{\theta_1-1}\chi_u\right) \\ &= \left(\sum_{u=0}^{q-1}\mu_{0,u}, \sum_{u=0}^{q-1}\mu_{1,u}, \ldots, \sum_{u=0}^{q-1}\mu_{q-1,u}\right), \end{aligned}$$

където $(\mu_{j,0}, \ldots, \mu_{j,q-1})$ е първият ред на матрицата $M_{k-1}^{[\chi^{(j)}]}$, $0 \le j \le q-1$, $j \neq 1$, а $(\mu_{1,0}, \ldots, \mu_{1,q-1})$ е първият ред на обновената чрез Add0 подмат-

рица $M_{k-1}^{[\chi^{(1)}]}$.

3. AllRows: Чрез ядрото на това преобразуване се изчисляват q реда AllRows$[j]$, $j = 0, 1, \ldots, q-1$, от матрицата $M_k^{[\chi]}$. За целта се ползва помощният масив TEMP с размери $q \times q$. Векторите AllRows$[j]$ се изчисляват от TEMP по формулите

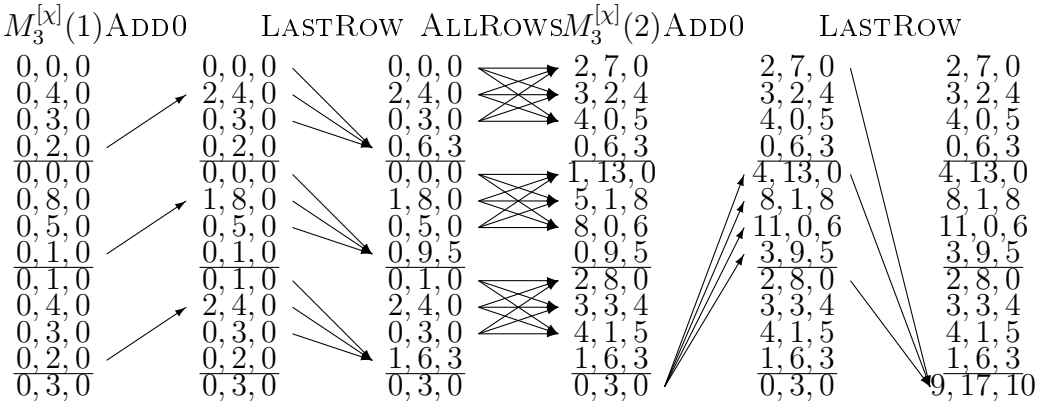
$$\begin{aligned} \text{AllRows}[0](\text{TEMP}) &= \text{TEMP}[0]+\text{TEMP}[1]+\cdots+\text{TEMP}[q-1], \\ \text{AllRows}[j](\text{TEMP}) &= \text{TEMP}[0]+SR_{\alpha_j}(\text{TEMP}[1])+\cdots+ \\ &\quad SR_{\alpha_j\alpha_{q-1}}(\text{TEMP}[q-1]),\ j>0, \end{aligned}$$

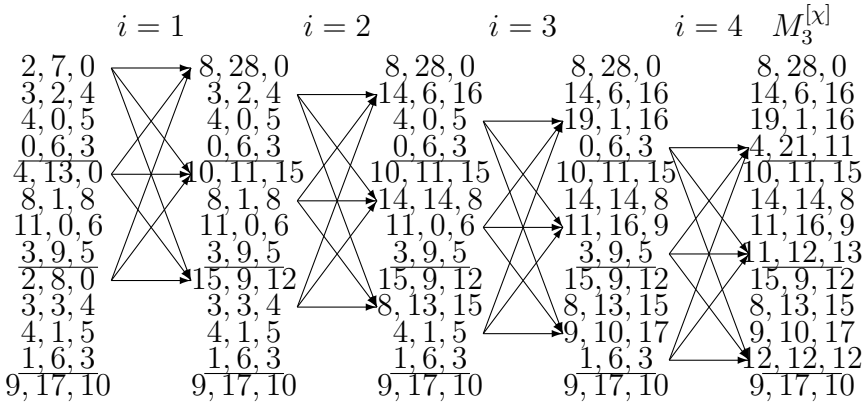
където TEMP[0], TEMP[1], ..., TEMP$[q-1]$ са редовете на TEMP.

В началото TEMP се състои от първите редове на всички подматрици $M_{k-1}^{[\chi^{(j)}]}$, $j = 0, 1, \ldots, q-1$, които са част от $M_k^{[\chi]}(k-1)$. Стойностите на AllRows$[j]$ обновяват съответните редове в $M_k^{[\chi]} = M_k^{[\chi]}(k)$. Тази процедура се повтаря $\theta(q, k-1)$ пъти, като на i-та итерация TEMP се състои от i-тите редове на горните подматрици и се прави съответното обновяване в $M_k^{[\chi]}$. По този начин се пресмятат всички редове на матрицата без последния.

В алгоритъма, при пресмятането на $M_k^{[\chi]}(l)$ от $M_k^{[\chi]}(l-1)$, се запазват неактивните редове непроменени и се прилагат гореописаните преобразувания за получаване на матрицата $M_l^{[\chi']}(l)$ от матрицата $M_l^{[\chi']}(l-1)$, където χ' е подходяща част от χ.

Пример 2.11. В схемите по-долу е илюстрирано прилагането на Алгоритъм 2 и Алгоритъм 3 за $q = 3$, $k = 3$ и $\chi = (0, 4, 3, 2, 0, 8, 5, 1, 1, 4, 3, 2, 3)$.





За да се обясни по-формално основният алгоритъм, може да се въведе матрично представяне на стъпките на преобразуванията между частичните характеристични разпределения.

Нека всички редове на матрицата $M_k^{[\chi]}(l)$ са наредени в един вектор с дължина $q\theta(q,k)$, който ще бъде означаван с $\widetilde{M}_k^{[\chi]}(l)$, $l=1,\ldots,k$. За краткост по-долу се ползват означенията: $\widetilde{M}_k^{[\chi]} = \widetilde{M}_k^{[\chi]}(k)$ и $\widetilde{\chi} = \widetilde{M}_k^{[\chi]}(1)$.

В следващата теорема се ползват матрици от няколко типа, а именно:

- Единичните $s \times s$ матрици I_s.
- $q \times q$ пермутационни матрици P_j, които осъществяват съответно пермутациите SR_{α_j}. В частност, $P_0 = I_q$, $P_1 = \begin{pmatrix} \mathbf{0} & 1 \\ I_{q-1} & \mathbf{0}^{\mathrm{T}} \end{pmatrix}$ и $P_j = P_1^j$.
- $q \times q$ матрици E_j, $j = 0, 1, \ldots, q-1$, за които $j+1$-ият ред на E_j се състои само от единици, а останалите редове на матрицата се състоят от нули.
- Матрици O, състоящи се само от нули, с размери по подразбиране.
- Матриците $T_{k,l}$ за $k, l \in \mathbb{Z}$, $2 \le l \le k$, които се дефинират индуктивно по следните правила:

  1. ако $k = l = 2$, то

$$T_{2,2} = \begin{pmatrix} I_q & I_q & I_q & \ldots & I_q & P_1^{-1} \\ I_q & P_1 & P_{\alpha_2} & \ldots & P_{\alpha_{q-1}} & I_q \\ I_q & P_{\alpha_2} & P_{\alpha_2^2} & \ldots & P_{\alpha_2\alpha_{q-1}} & P_{\alpha_2} \cdot P_1^{-1} \\ \vdots & & & & & \\ I_q & P_{\alpha_{q-1}} & P_{\alpha_{q-1}\alpha_2} & \ldots & P_{\alpha_{q-1}^2} & P_{\alpha_{q-1}} \cdot P_1^{-1} \\ E_0 & E_1 & E_2 & \ldots & E_{q-1} & E_1 \end{pmatrix}; \qquad (2.8)$$

2. ако $k > l$, то

$$T_{k,l} = \begin{pmatrix} I_q \otimes T_{k-1,l} & O \\ O & I_q \end{pmatrix}; \tag{2.9}$$

3. ако $k = l > 2$, то

$$T_{k,k} = \begin{pmatrix} I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes I_q & \dots & I_{\theta(q,k-1)} \otimes I_q & \mathbf{1}^{\mathrm{T}} \otimes P_1^{-1} \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_1 & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}} & \mathbf{1}^{\mathrm{T}} \otimes I_q \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_{\alpha_2} & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_2\alpha_{q-1}} & \mathbf{1}^{\mathrm{T}} \otimes P_2 \cdot P_1^{-1} \\ \vdots & & & & \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}} & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}^2} & \mathbf{1}^{\mathrm{T}} \otimes P_{\alpha_{q-1}} \cdot P_1^{-1} \\ E_0 \;\; O & E_1 \;\; O & \dots & E_{q-1} \;\; O & I_q \end{pmatrix}. \tag{2.10}$$

Теорема 2.5. *Нека χ е характеристичен вектор на линеен $[n,k]_q$ код. Тогава*

$$\left(\widetilde{M}_k^{[\chi]}(l)\right)^{\mathrm{T}} = T_{k,l} \cdot \left(\widetilde{M}_k^{[\chi]}(l-1)\right)^{\mathrm{T}}, \quad l = 2, \dots, k, \tag{2.11}$$

и

$$\left(\widetilde{M}_k^{[\chi]}\right)^{\mathrm{T}} = T_{k,k} \cdot T_{k,k-1} \cdots T_{k,2} \cdot \widetilde{\chi}^{\mathrm{T}}. \tag{2.12}$$

Доказателство. Нека $k = 2$. Тогава $\theta(q,2) = q+1$, M_2 е $(q+1)\times(q+1)$ матрица и характеристичният вектор χ има дължина $q+1$. Нека $\chi = (\chi_0, \chi_1, \dots, \chi_q)$. За получаване на $M_2^{[\chi]}(2)$ се прилагат преобразуванията Add0, LastRow и AllRows върху матрицата $M_2^{[\chi]}(1) = \begin{pmatrix} M_1^{[\chi_0]} \\ \vdots \\ M_1^{[\chi_{q-1}]} \\ M_1^{[\chi_q]} \end{pmatrix}$ (съгласно дефиниция 2.5).

Трите преобразувания имат матрично представяне, като съответните матрици са квадратни от ред $q(q+1)$ и са съответно

$$T_0 = \begin{pmatrix} I_q & O & \cdots & O & O \\ O & I_q & \cdots & O & P_1^{-1} \\ & & \ddots & & \\ O & O & \cdots & I_q & O \\ O & O & \cdots & O & I_q \end{pmatrix}, \quad T_{last} = \begin{pmatrix} I_q & O & \cdots & O & O \\ O & I_q & \cdots & O & O \\ & & \ddots & & \\ O & O & \cdots & I_q & O \\ E_0 & E_1 & \cdots & E_{q-1} & O \end{pmatrix},$$

$$T_{all} = \begin{pmatrix} I_q & I_q & I_q & \cdots & I_q & O \\ I_q & P_1 & P_{\alpha_2} & \cdots & P_{\alpha_{q-1}} & O \\ I_q & P_{\alpha_2} & P_{\alpha_2^2} & \cdots & P_{\alpha_2\alpha_{q-1}} & O \\ & & & \ddots & & \\ I_q & P_{\alpha_{q-1}} & P_{\alpha_{q-1}\alpha_2} & \cdots & P_{\alpha_{q-1}^2} & O \\ O & O & O & \cdots & O & I_q \end{pmatrix}. \tag{2.13}$$

Матрицата $T_{2,2}$ е произведение на горните матрици:

$$T_{2,2} = T_{all} \cdot T_{last} \cdot T_0 = \begin{pmatrix} I_q & I_q & I_q & \ldots & I_q & P_1^{-1} \\ I_q & P_1 & P_{\alpha_2} & \ldots & P_{\alpha_{q-1}} & I_q \\ I_q & P_{\alpha_2} & P_{\alpha_2^2} & \ldots & P_{\alpha_2\alpha_{q-1}} & P_{\alpha_2} \cdot P_1^{-1} \\ \vdots & & & & & \\ I_q & P_{\alpha_{q-1}} & P_{\alpha_{q-1}\alpha_2} & \ldots & P_{\alpha_{q-1}^2} & P_{\alpha_{q-1}} \cdot P_1^{-1} \\ E_0 & E_1 & E_2 & \ldots & E_{q-1} & E_1 \end{pmatrix}. \tag{2.14}$$

Проверката на равенството $(\widetilde{M}_2^{[\chi]})^{\mathrm{T}} = T_{2,2} \cdot (\widetilde{M}_2^{[\chi]}(1))^{\mathrm{T}}$ е непосредствена.

Нека $k > 2$ и твърдението на теоремата е изпълнено за всяко $k' \in \mathbb{Z}$, за което $2 \leq k' < k$. Характеристичният вектор $\chi \in \mathbb{Z}^{\theta(q,k)}$ може да бъде разделен на $q+1$ части съгласно (2.5).

Ако $k > l$, то

$$M_k^{[\chi]}(l) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]}(l) \\ M_{k-1}^{[\chi^{(1)}]}(l) \\ \ldots \\ M_{k-1}^{[\chi^{(q-1)}]}(l) \\ M_1^{[\chi^{(q)}]} \end{pmatrix} \quad \text{и} \quad M_k^{[\chi]}(l-1) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]}(l-1) \\ M_{k-1}^{[\chi^{(1)}]}(l-1) \\ \ldots \\ M_{k-1}^{[\chi^{(q-1)}]}(l-1) \\ M_1^{[\chi^{(q)}]} \end{pmatrix}.$$

Съгласно индукционното предположение

$$\begin{aligned} (\widetilde{M}_{k-1}^{[\chi^{(0)}]}(l))^{\mathrm{T}} &= T_{k-1,l} \cdot (\widetilde{M}_{k-1}^{[\chi^{(0)}]}(l-1))^{\mathrm{T}}, \\ (\widetilde{M}_{k-1}^{[\chi^{(1)}]}(l))^{\mathrm{T}} &= T_{k-1,l} \cdot (\widetilde{M}_{k-1}^{[\chi^{(1)}]}(l-1))^{\mathrm{T}}, \\ &\ldots \\ (\widetilde{M}_{k-1}^{[\chi^{(q-1)}]}(l))^{\mathrm{T}} &= T_{k-1,l} \cdot (\widetilde{M}_{k-1}^{[\chi^{(q-1)}]}(l-1))^{\mathrm{T}}. \end{aligned}$$

Оттук твърдението следва директно.

Ако $k = l$, е изпълнено

$$M_k^{[\chi]}(k) = M_k^{[\chi]} \quad \text{и} \quad M_k^{[\chi]}(k-1) = \begin{pmatrix} M_{k-1}^{[\chi^{(0)}]} \\ M_{k-1}^{[\chi^{(1)}]} \\ \dots \\ M_{k-1}^{[\chi^{(q-1)}]} \\ M_1^{[\chi^{(q)}]} \end{pmatrix}.$$

Прилагането на (2.6) дава

$$\widetilde{M}_k^{[\chi]} = T_{k,k} \cdot \left(\widetilde{M}_{k-1}^{[\chi^{(0)}]} | \widetilde{M}_{k-1}^{[\chi^{(1)}]} | \dots | \widetilde{M}_{k-1}^{[\chi^{(q-1)}]} | \widetilde{M}_1^{[\chi^{(q)}]}\right)^{\mathrm{T}} = T_{k,k} \cdot \widetilde{M}_k^{[\chi]}(k-1).$$

С горните разсъждения, основното твърдение е доказано. □

2.4 Съкратено характеристично разпределение

Характеристичното разпределение на вектор b по отношение на характеристичен вектор на линеен код е вектор с дължина q и сума от координатите, равна на дължината на кода. Достатъчно е да се знаят $q-1$ от тези координати, за да се получи последната. Това дава основание за въвеждане на следващото понятие.

Дефиниция 2.6. Нека $\chi \in \mathbb{Z}^t$ и $b \in \mathbb{F}_q^t$, $t \in \mathbb{N}$. *Съкратеното характеристично разпределение* на вектора b по отношение на χ е векторът

$$b^{[\chi]_r} = (\mu_0 - \mu_1, \dots, \mu_0 - \mu_{q-1}) \in \mathbb{Z}^{q-1},$$

където $b^{[\chi]} = (\mu_0, \mu_1, \dots, \mu_{q-1})$ е характеристичното разпределение на b по отношение на χ.

Лема 2.6. *Ако $\chi \in \mathbb{Z}^t$ и $b \in \mathbb{F}_q^t$, $t \in \mathbb{N}$, то*

$$(b^{[\chi]_r})^{\mathrm{T}} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & -1 & 0 \\ 1 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix} \cdot (b^{[\chi]})^{\mathrm{T}}.$$

Доказателството следва непосредствено от дефиницията.

Твърдение 2.7. *Нека* $\chi = (\chi_1, \ldots, \chi_t) \in \mathbb{Z}^t$ *и* $b = (b_1, \ldots, b_t) \in \mathbb{F}_q^t$, $t \in \mathbb{N}$. *Тогава свкратеното характеристично разпределение* $b^{[\chi]_r}$ *на вектора* b *по отношение на* χ *има следните свойства:*

1. *Сумата от координатите на* $b^{[\chi]_r}$ *е*

$$(q-1)\mu_0 - \mu_1 - \ldots - \mu_{q-1} = q\mu_0 - \sum_{j=0}^{q-1} \mu_j = q\mu_0 - \sum_{u=1}^{t} \chi_u.$$

2. *Ако* $b = \boldsymbol{\alpha_j}$, $j \neq 0$, *свкратеното характеристично разпределение* $b^{[\chi]_r}$ *се свстои от нули с изключение на* j*-ия елемент, който е равен на* $-\sum_{u=1}^{t} \chi_u$. *Така например*

$$\mathbf{1}^{[\chi]_r} = \left(-\sum_{i=1}^{t} \chi_i, 0 \ldots, 0 \right).$$

За $b = \mathbf{0}$ *е изпълнено* $\mathbf{0}^{[\chi]_r} = \left(\sum_{i=1}^{t} \chi_i, \sum_{i=1}^{t} \chi_i, \ldots, \sum_{i=1}^{t} \chi_i \right)$.

3. *Ако* $\mathcal{N}(b)$ *се получава от* b *чрез заместване на всички ненулеви координати с* 1, *то*

$$\mathcal{N}(b) \cdot \chi^{\mathrm{T}} = (q-1)\mu_0 - b^{[\chi]_r} \cdot \mathbf{1}^{\mathrm{T}}.$$

4. *Ако към всяка координата на* b *се добави* 1 *(над* $\mathbb{F}_q$*), свкратеното характеристично разпределение ще се промени, както следва:*
$b^{[\chi]_r} = (\mu_0 - \mu_1, \ldots, \mu_0 - \mu_{q-1})$
$\Longrightarrow (b + \mathbf{1})^{[\chi]_r} = (\mu_{q-1} - \mu_0, \mu_{q-1} - \mu_1, \ldots, \mu_{q-1} - \mu_{q-2})$.
Оказва се, че $\left((b + \mathbf{1})^{[\chi]_r}\right)^{\mathrm{T}} = R_1 \cdot \left(b^{[\chi]_r}\right)^{\mathrm{T}}$, *където*

$$R_1 = \begin{pmatrix} 0 & 0 & \ldots & 0 & 0 & -1 \\ 1 & 0 & & 0 & 0 & -1 \\ 0 & 1 & & 0 & 0 & -1 \\ \vdots & & & & & \\ 0 & 0 & & 1 & 0 & -1 \\ 0 & 0 & & 0 & 1 & -1 \end{pmatrix}$$

5. *Прибавянето на елемента* $j \in \mathbb{F}_q$ *към всяка координата на* b *променя свкратеното характеристично разпределение така* $((b + \boldsymbol{j})^{[\chi]_r})^{\mathrm{T}} = R_1^j \cdot (b^{[\chi]_r})^{\mathrm{T}}$.

6. *Ако* $s, t \in \mathbb{N}$, $\chi' \in \mathbb{Z}^s$, $\chi'' \in \mathbb{Z}^t$, $b' \in \mathbb{F}_q^s$, $b'' \in \mathbb{F}_q^t$, $\chi = (\chi'|\chi'')$, $b = (b'|b'')$, *то*

$$b^{[\chi]_r} = b'^{[\chi']_r} + b''^{[\chi'']_r}.$$

Горните свойства следват непосредствено от твърдение 2.2 и дефиниция 2.6.

Аналогично на описаното в предишния раздел се въвеждат понятията съкратено характеристично разпределение на матрица, частично съкратено характеристично разпределение и неговото представяне като вектор.

Следващата лема обобщава първото свойство от твърдение 2.7.

Лема 2.8. *Нека* χ *е характеристичен вектор на линеен* $[n,k]_q$ *код с пълна дължина. Тогава сумата от координатите на* $\widetilde{M}_k^{[\chi]_r}$ *е* $-n$.

Доказателство. Тъй като χ е характеристичен вектор на линеен $[n,k]_q$ код с пълна дължина, то $\sum_{u=1}^{\theta(q,k)} \chi_u = n$. Нека c_i е i-ият ред на M_k, $i = 1, \ldots, \theta(q,k)$, а $c_i^{[\chi]} = (\mu_0, \mu_1, \ldots, \mu_{q-1})$. Съгласно свойство 1 от твърдение 2.7 сумата от координатите на $c_i^{[\chi]_r}$ е равна на $q\mu_0 - \sum_{u=1}^{\theta(q,k)} \chi_u = q\mu_0 - n = q(\mu_0 - n) + (q-1)n = -q\mathrm{wt}(c_i) + (q-1)n$. Последното равенство следва от лема 2.1 и теорема 2.4. Следователно сумата от координатите на $\widetilde{M}_k^{[\chi]_r}$ е равна на $-q\sum_{i=1}^{\theta(q,k)} \mathrm{wt}(c_i) + (q-1)n\theta(q,k)$.

Така остава да се пресметне сумата $\sum_{i=1}^{\theta(q,k)} \mathrm{wt}(c_i)$ от теглата по Хеминг на кодовите думи c_i, $i = 1, \ldots, \theta(q,k)$, които формират максимално подмножество U от непропорционални кодови думи на разглеждания линеен код. Така сумата на тези тегла е равна на $\frac{1}{q-1}\sum_{w=1}^{n} wA_w = nq^{k-1}$, където A_w е броят на кодовите думи с тегло w, $w = 1, \ldots, n$ (тези суми се наричат степенни моменти на Плес [49]). Оттук следва, че

$$-q\sum_{i=1}^{\theta(q,k)} \mathrm{wt}(c_i) + (q-1)n\theta(q,k) = -nq^k + (q-1)n\frac{q^k-1}{q-1} = -nq^k + nq^k - n = -n.$$

□

2.5 Сложност на алгоритмите и експериментални резултати

Разгледани са кодове над просто поле $\mathbb{F}_q$ с дължина $n < 2^{32}$ и брой на кодовите думи $q^k < 2^{64}$, така че да са нужни променливи от тип 32-bit integers за

представяне на теглата на кодовите думи и 64-bit integers за броя на кодовите думи с дадено тегло. Така се ползват само променливи от основния тип integer и операции с тях. За пресмятане на тегловното разпределение на линеен код са използвани два масива с променливи от тип 32-bit integers, именно D с размери $\theta(q,k) \times q$ и TEMP с размери $q \times q$. Общо използваната памет, която е нужна (без да се брои пораждащата матрица) е $q\theta(q,k) + q^2 + 2n + A$ променливи от тип 32-bit integers. Тук е добавено $2n$, тъй като тегловното разпределение е вектор с дължина n с координати от тип 64-bit integers, както и константа A за други променливи, които алгоритъмът използва.

Основният алгоритъм пресмята масива D на $k-1$ стъпки. На стъпка l има $q^{k-l}\theta(q,l)$ активни и $\theta(q,k-l)$ неактивни реда, $l = 2,3,\ldots,k$. Неактивните редове остават непроменени. За преобразуването ADD0 в Алгоритъм 3 се ползват $q^{k-l}\theta(q,l-1) \leq \theta(q,k-1)$ операции. Чрез преобразуванията LastRow или AllRows един елемент от активен ред се изчислява по Алгоритъм 3 като сума на q събираеми и следователно се извършват още $q^2q^{k-l}\theta(q,l)$ операции. Окончателно, сложността на стъпка l (в тялото на for цикъла в Алгоритъм 2) е

$$\begin{aligned} q^2q^{k-l}\theta(q,l) + q^{k-l}\theta(q,l-1) &= q^{k+2-l}\frac{q^l-1}{q-1} + q^{k-l}\frac{q^{l-1}-1}{q-1} \\ &= \frac{q^{k+2} - q^{k+2-l} + q^{k-1} - q^{k-l}}{q-1}. \end{aligned}$$

Оттук общата сложност на Алгоритъм 2 е

$$\sum_{l=2}^{k} \frac{q^{k+2} - q^{k+2-l} + q^{k-1} - q^{k-l}}{q-1} = (k-1)\frac{q^{k+2} + q^{k-1}}{q-1} - \frac{(q^2+1)(q^{k-1}-1)}{(q-1)^2}.$$

Това дава, че за фиксирано q сложността на алгоритъма е $O(kq^k)$. Когато се считат k и q за променливи, времето за изпълнение е $O(kq^{k+1})$.

Забележка 2.1. Бе направено сравнение на предложения алгоритъм с Algorithm 9.8 (Walsh transform over a prime finite field $\mathbb{F}_p$) в [50]. Според Жу, сложността на този алгоритъм, когато p се променя, е $O(kp^{k+2})$.

Представеният метод е програмиран на C/C++ [22]. За да се сравни ефективността, е ползвана програма на C, в която е вложен алгоритъмът, описан в [23], чиято сложност е като на алгоритмите, базирани на кодове на Грей. За среда на разработка на двете програми е ползвано MS Visual Studio 2012. Всички примерни данни са изчислявани с програмите върху следната платформа: Intel Core i7-3770k 3.50 GHz processor in Active solution configuration — Release, and Active solution platform — X64.

Входни данни са случайно генерирани линейни кодове с дължини 30, 300, 3000, 30000 и различни размерности над полета с 2, 3, 4, 5 и 7 елемента. Всички резултати с времето за изпълнение в секунди са дадени в Таблица 2.1. В колоните 'NEW' е дадено времето за изпълнение по новия алгоритъм (описан в тази глава), а в колоните 'OLD' е дадано времето за изпълнение при същите входни данни, но от програмата по алгоритъма, описан в [23], включена в пакета Q-Extension. Времето за изпълнение, показано в Таблица 2.1, е пълното време за изчисляване на тегловното разпределение при вход пораждащата матрица на кода.

В Таблица 2.2 са представени резултати за същите параметри, както в Таблица 2.1, но получени с Magma V2.25-2 онлайн чрез Magma Calculator, стартиран на виртуална машина с Intel Xeon Processor E3-1220, 3.10 GHz.

Резултатите, дадени в таблиците, показват, че представеният метод е по-бърз за кодове с големи дължини. Времето за изчисляване на характеристичния вектор е пренебрежимо малко.

Коментари към Глава 2

През 2015 г. проф. Илия Буюклиев и проф. Стефка Буюклиева се запознават с работата на Марк Карповски [51] за прилагане на трансформацията на Уолш-Адамар за намиране на тегловно разпределение на двоични линейни кодове. Tatsuya Maruta предлага обобщение на метода за линейни кодове над крайни полета.

През 2016 г. започва съвместната работа на четиримата съавтори, чиито резултати са описани в тази глава и са публикувани през 2021 г. [P3]. Резултатите са поетапно докладвани [D1, D2, D4, D6, D7, D11]. Реализирането на алгоритмите чрез програма на C/C++ и описването на експерименталните резултати е дело на проф. Илия Буюклиев.

Таблица 2.1: Експериментални резултати

		$n =$	30	$n =$	300	$n =$	3000	$n =$	30000
q	k	NEW	OLD	NEW	OLD	NEW	OLD	NEW	OLD
2	23	0.190	0.084	0.132	0.139	0.131	0.700	0.133	6.064
2	24	0.214	0.168	0.268	0.278	0.266	1.412	0.271	11.665
2	25	0.552	0.337	0.552	0.549	0.552	2.816	0.552	23.649
2	26	1.146	0.637	1.144	1.107	1.148	5.595	1.150	47.001
3	13	0.039	0.015	0.101	0.032	0.101	0.190	0.103	1.690
3	14	0.292	0.048	0.295	0.099	0.294	0.575	0.295	5.070
3	15	0.968	0.145	0.949	0.291	0.955	1.716	0.957	15.122
3	16	3.012	0.439	3.035	0.881	3.100	5.249	3.111	46.695
4	10	0.016	0.007	0.016	0.013	0.016	0.089	0.016	0.747
4	11	0.064	0.025	0.064	0.052	0.065	0.344	0.066	2.997
4	12	0.261	0.109	0.263	0.208	0.263	1.325	0.263	11.637
4	13	1.444	0.422	1.444	0.857	1.445	5.359	1.446	46.976
5	8	0.130	0.013	0.140	0.133	0.140	1.225	0.150	12.228
5	9	0.063	0.068	0.063	0.614	0.065	6.135	0.067	60.834
5	10	0.335	0.309	0.335	3.062	0.337	30.318	0.343	295.691
5	11	1.847	1.517	1.842	15.197	1.841	151.716	1.843	1514.924
7	6	0.004	0.002	0.004	0.020	0.004	0.202	0.008	2.049
7	7	0.027	0.013	0.022	0.166	0.021	1.469	0.026	14.554
7	8	0.170	0.107	0.174	1.037	0.172	10.084	0.181	101.280
7	9	1.351	0.743	1.363	7.105	1.379	70.795	1.397	705.218

Таблица 2.2: Експериментални резултати Magma V2.25-2

q	k	n= 30	n= 300	n= 3000	n= 30000
2	23	0.000	0.130	1.140	11.340
2	24	0.000	0.260	2.300	22.680
2	25	0.000	0.520	4.560	45.380
2	26	0.000	1.030	9.150	90.680
3	13	0.010	0.040	0.180	1.730
3	14	0.010	0.080	0.540	5.180
3	15	0.060	0.240	1.580	15.530
3	16	0.020	0.630	4.810	46.580
4	10	0.010	0.040	0.090	0.730
4	11	0.010	0.070	0.320	2.940
4	12	0.060	0.180	1.250	11.760
4	13	0.230	0.680	5.000	47.030
5	8	0.00	0.070	0.040	0.330
5	9	0.00	0.090	0.170	1.670
5	10	0.03	0.170	0.860	8.350
5	11	0.160	0.600	4.270	41.770
7	6	0.00	0.00	0.010	0.080
7	7	0.00	0.010	0.060	0.530
7	8	0.010	0.140	0.370	3.600
7	9	0.100	0.410	2.630	25.300

Глава 3

Методи за пресмятане на тегловно разпределение на линеен код над съставно крайно поле

3.1 Подход чрез код на следите

Нека $q = p^m$, където p е просто число и $m > 1$.

Ако матрицата G е пораждаща за линейния $[n,k]_q$ код C, то *разширената матрица* $\overline{G} = (\alpha_1 G|\alpha_2 G|\cdots|\alpha_{q-1}G)$ е пораждаща за линеен $[(q-1)n,k]_q$ код $\overline{C}$. Ако минималното тегло на C е d, то минималното тегло на $\overline{C}$ е $(q-1)d$.

За всеки вектор $x \in \mathbb{F}_q^n$ нека $\mathrm{Tr}(x) = (\mathrm{Tr}(x_1), \ldots, \mathrm{Tr}(x_n)) \in \mathbb{F}_p^n$.

Дефиниция 3.1. Нека C е линеен $[n,k]_q$ код с пораждаща матрица G. Кодът $\mathrm{Tr}(C) = \{\mathrm{Tr}(c)|c \in C\}$ се нарича *код на следите* на C.

$\mathrm{Tr}(C)$ е линеен код над простото поле $\mathbb{F}_p$ със същата дължина като C, но с размерност по-малка или равна на mk [68]. Затова вместо $\mathrm{Tr}(C)$ ще бъде разгледан кодът на следите на $\overline{C}$.

Лема 3.1. *Размерността на кода* $\mathrm{Tr}(\overline{C})$ *е равна на* mk.

Доказателство. Ако $c_1, \ldots, c_k$ и $\bar{c}_1, \ldots, \bar{c}_k$ са съответно редовете на G и $\overline{G}$, то $\bar{c}_t = (\alpha_1 c_t|\alpha_2 c_t|\cdots|\alpha_{q-1}c_t)$, $t = 1, \ldots, k$. Нека $\beta_1, \beta_2, \ldots, \beta_m$ е базис на $\mathbb{F}_q$ над $\mathbb{F}_p$. Ще бъде доказано, че векторите $\mathrm{Tr}(\beta_s \bar{c}_t)$, $s = 1, \ldots, m$, $t = 1, \ldots, k$, образуват базис на $\mathrm{Tr}(\overline{C})$.

Нека числата $\lambda_{st} \in \mathbb{F}_p$, $s = 1, \ldots, m$, $t = 1, \ldots, k$, са такива, че

$$\sum_{s=1}^{m}\sum_{t=1}^{k} \lambda_{st}\mathrm{Tr}(\beta_s \bar{c}_t) = \mathbf{0}.$$

Като части от този вектор

$$\sum_{s=1}^{m}\sum_{t=1}^{k}\lambda_{st}\mathrm{Tr}(\alpha_j\beta_s c_t)=\mathrm{Tr}\left(\alpha_j\sum_{s=1}^{m}\sum_{t=1}^{k}\lambda_{st}\beta_s c_t\right)=\mathbf{0},\quad j=1,2,\ldots,q-1.$$

Ако някоя координата на вектора $\sum_{s=1}^{m}\sum_{t=1}^{k}\lambda_{st}\beta_s c_t$ е различна от 0, то съответните координати на векторите $\alpha_j\sum_{s=1}^{m}\sum_{t=1}^{k}\lambda_{st}\beta_s c_t$, $j=1,2,\ldots,q-1$, са всичките ненулеви елементи на $\mathbb{F}_q$. Следователно някои от техните следи са ненулеви елементи на $\mathbb{F}_p$, което е противоречие. Това доказва, че $\sum_{s=1}^{m}\sum_{t=1}^{k}\lambda_{st}\beta_s c_t=\mathbf{0}$. Тъй като $c_1,\ldots,c_k$ е базис на C, то $\sum_{s=1}^{m}\lambda_{st}\beta_s=0$, за всяко $t=1,2,\ldots,k$. Но $\beta_1,\beta_2,\ldots,\beta_m$ е базис на $\mathbb{F}_q$ над $\mathbb{F}_p$, откъдето $\lambda_{st}=0$ за всяко $s=1,\ldots,m$, $t=1,\ldots,k$. Следователно векторите $\mathrm{Tr}(\beta_s\bar{c}_t)$, $s=1,\ldots,m$, $t=1,\ldots,k$, са линейно независими и размерността на $\mathrm{Tr}(\overline{C})$ е mk. □

Следствие 3.2. *Кодовете C и $\mathrm{Tr}(\overline{C})$ имат един и същ брой кодови думи, именно $q^k=p^{mk}$.*

Теорема 3.3. *Нека $q=p^m$, където p е просто число и $m>1$. Нека C е линеен $[n,k]_q$ код с тегловна функция $W(z)=\sum_{w=0}^{n}A_w z^w$. Тогава $\mathrm{Tr}(\overline{C})$ е линеен $[(q-1)n,mk]_p$ код с тегловна функция*

$$W_1(z)=\sum_{w=0}^{n}A_w z^{\frac{q(p-1)w}{p}}.$$

Доказателство. Нека $c\in C$ и $\bar{c}=\mathrm{Tr}(\alpha_1 c|\alpha_2 c|\ldots|\alpha_{q-1}c)$ е съответната кодова дума от $\overline{C}$. Нека някоя координата на c има стойност $a\neq 0$. Тогава $\{\alpha_1 a,\alpha_2 a,\ldots,\alpha_{q-1}a\}=\mathbb{F}_q\backslash\{0\}$. Понеже Tr е линейно изображение върху $\mathbb{F}_p$, ядрото му съдържа p^{m-1} елемента. Следователно p^m-p^{m-1} от елементите на множеството $\{\mathrm{Tr}(\alpha_1 a),\mathrm{Tr}(\alpha_2 a),\ldots,\mathrm{Tr}(\alpha_{q-1}a)\}$ са различни от 0. Оттук

$$\mathrm{wt}(\bar{c})=(p^m-p^{m-1})\mathrm{wt}(c)=\frac{q(p-1)}{p}\mathrm{wt}(c).$$

Сега твърдението следва непосредствено. □

Съгласно горната теорема тегловното разпределение на линеен код C над съставно крайно поле може да се получи от тегловното разпределение

на линейния код $\mathrm{Tr}(\overline{C})$, който е над просто поле, като се приложи алгоритъма, описан в предходната глава. Сложността за изчисляване на характеристичния вектор на $\mathrm{Tr}(\overline{C})$ е $O(mkqn)$, а на характеристичното разпределение – $O(mkp^{mk+1}) = O(kmpq^k)$.

3.2 Подход чрез трансформация на следите

Разсъжденията до края на тази глава са направени за съставно крайно поле с характеристика 2. Те с лекота могат да се обобщят при друга характеристика на разглежданото съставно крайно поле.

Нека $q = 2^m$ и $\beta_1, \ldots, \beta_m$ е самодуален базис на $\mathbb{F}_{2^m}$ над $\mathbb{F}_2$. Съгласно теорема 1.1 такъв базис съществува. Нека с $\lambda(\alpha) = (\lambda_1(\alpha), \ldots, \lambda_m(\alpha))$ е означен векторът $\lambda(\alpha) \in \mathbb{F}_2^m$, съответстващ на елемента

$$\alpha = \lambda_1(\alpha)\beta_1 + \cdots + \lambda_m(\alpha)\beta_m \in \mathbb{F}_q.$$

До края тази глава, нека елементите $\alpha_0 = 0, \alpha_1, \ldots, \alpha_{q-1}$ на $\mathbb{F}_q$ са наредени така, че съответните двоични вектори $\lambda(0), \lambda(\alpha_1), \ldots, \lambda(\alpha_{q-1})$ са наредени лексикографски.

Нека G е пораждаща матрица на линеен $[n,k]_q$ код C с пълна дължина, където $q = 2^m$. Нека f_G е характеристичната функция на C съгласно дефиниция 1.20. Теорема 1.11 дава връзката между тегловното разпределение на C и трансформацията на следите на f_G, която по дефиниция 1.19 е функцията

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_q^k} f_G(x)\tau_\omega(x) = \sum_{x \in \mathbb{F}_q^k} f_G(x)(-1)^{\mathrm{Tr}(\langle \omega, x \rangle)}, \quad \omega \in \mathbb{F}_q^k. \tag{3.1}$$

Векторите от стойностите на $\widehat{f}$ и f_G са свързани с равенството $TT_{\widehat{f}} = T_k \cdot TT_{f_G}$, при което индексите $\omega, x \in \mathbb{F}_q^k$, определящи поредността съответно на редовете и стълбовете в матрицата $T_k = (\tau_\omega(x))$, са наредени лексикографски.

Лема 3.4. *Матрицата*

$$T_1 = \left((-1)^{\mathrm{Tr}(\alpha_j \alpha_{j'})}\right)_{j,j'=0}^{q-1}$$

е трансформационната матрица H_m, дефинирана с (1.3).

Доказателство. От линейността на следата над $\mathbb{F}_2$ за $j, j' = 0, 1, \ldots, q-1$ е изпълнено

$$\begin{aligned}
\mathrm{Tr}(\alpha_j \alpha_{j'}) &= \mathrm{Tr}\left(\sum_{s=1}^{m} \lambda_s(\alpha_j)\beta_s \sum_{s'=1}^{m} \lambda_{s'}(\alpha_{j'})\beta_{s'}\right) \\
&= \mathrm{Tr}\left(\sum_{s=1}^{m}\sum_{s'=1}^{m} \lambda_s(\alpha_j)\lambda_{s'}(\alpha_{j'})\beta_s\beta_{s'}\right) \\
&= \sum_{s=1}^{m}\sum_{s'=1}^{m} \mathrm{Tr}\left(\lambda_s(\alpha_j)\lambda_{s'}(\alpha_{j'})\beta_s\beta_{s'}\right) \\
&= \sum_{s=1}^{m}\sum_{s'=1}^{m} \lambda_s(\alpha_j)\lambda_{s'}(\alpha_{j'})\mathrm{Tr}\left(\beta_s\beta_{s'}\right).
\end{aligned}$$

Тъй като базисът $\beta_1, \ldots, \beta_m$ е самодуален, то съгласно дефиниция 1.2

$$\lambda_s(\alpha_j)\lambda_{s'}(\alpha_{j'})\mathrm{Tr}\left(\beta_s\beta_{s'}\right) = \begin{cases} \lambda_s(\alpha_j)\lambda_s(\alpha_{j'}), & \text{ако } s = s', \\ 0 \qquad\qquad\qquad\quad , & \text{ако } s \neq s'. \end{cases}$$

Така

$$\mathrm{Tr}(\alpha_j\alpha_{j'}) = \sum_{s=1}^{m} \lambda_s(\alpha_j)\lambda_s(\alpha_{j'}) = \langle \lambda(\alpha_j), \lambda(\alpha_{j'}) \rangle.$$

Сега твърдението на теоремата следва от дефиниция 1.16 и теорема 1.7. □

Горната лема показва, че

$$T_1 = H_m = \otimes^m H_1 = \otimes^m \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.2}$$

Тъй като трансформационната матрица T_k е Кронекерова степен на T_1, то

$$T_k = \otimes^k T_1 = \otimes^{km} H_1 = \otimes^{km} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.3}$$

и може да се ползва бътерфлай алгоритъм за пресмятане на трансформацията (3.1). Псевдокод на този алгоритъм е описан в Алгоритъм 4. Подобен псевдокод е описан от Жу [50, Algorithm 9.3].

Пример 3.1. Нека $G = \begin{pmatrix} 1 & 0 & 1 & \beta_2 \\ 0 & 1 & \beta_2 & \beta_1 \end{pmatrix}$ е пораждаща матрица на линеен $[4, 2]_4$ код C над $\mathbb{F}_4 = \{0, 1, \beta_1, \beta_2\}$. Самодуалният базис β_1, β_2, където $\beta_2 =$

Алгоритъм 4 Бътерфлай алгоритъм за трансформацията на следите при поле с характеристика 2 и наредба на елементите по самодуален базис

Вход: Естествени числа k, m, $q = 2^m$ и масив f с дължина q^k — таблицата от стойностите на функцията $f_G(x)$

Изход: Обновен масивът f — таблица от стойностите на $\widehat{f}(\omega)$

```
1:  s = 1;
2:  for l = 1 to km do
3:     for t = 0 to 2^{km-l} - 1 do
4:        for i_0 = 0 to s - 1 do
5:           i_1 = 2st + i_0;
6:           A = f[i_1];
7:           B = f[i_1 + s];
8:           f[i_1] = A + B;
9:           f[i_1 + s] = A - B;
10:       end for
11:    end for
12:    s = 2s;
13: end for
```

$\beta_1 + 1 = \beta_1^2$, определя наредбата на елементите на полето $\alpha_0 = 0$, $\alpha_1 = \beta_2$, $\alpha_2 = \beta_1$, $\alpha_3 = 1$. Таблицата от стойностите на характеристичната функция f_G на кода C е $TT_{f_G} = (0,1,1,1,1,0,2,0,1,0,0,2,1,2,0,0)^{\mathrm{T}}$. След прилагане на Алгоритъм 4 се получава

$$TT_{\widehat{f}} = (12,0,0,0,0,-4,-4,4,0,4,-4,-4,0,-4,4,-4)^{\mathrm{T}}.$$

По (1.21) се извежда тегловната функция на кода C

$$W(z) = 1 + 3z^2 + 6z^3 + 6z^4.$$

3.3 Матрично представяне на подобрен алгоритъм

Усъвършенстване на пресмятанията може да се направи, когато на трансформация се подложи само част от вектора от стойностите на характеристичната функция, съответстваща на непропорционални стойности на аргумента. За целта е удобно да се ползват стълбовете на пораждащата матрица G_k на симплекс кода $\mathcal{S}_{q,k}$, индуктивно дефинирана по схемата в (2.1). Разработеният алгоритъм

ползва входни данни с дължина $\theta(q,k)$ вместо q^k. Сложността на алгоритъма е $O(mkq^{k-1})$.

Матриците G_k се дефинират индуктивно чрез равенствата

$$G_1 = (1), \quad G_k = \begin{pmatrix} \mathbf{0} & \boldsymbol{\alpha_1} & \dots & \boldsymbol{\alpha_{q-1}} & 1 \\ G_{k-1} & G_{k-1} & \dots & G_{k-1} & \mathbf{0}^{\mathrm{T}} \end{pmatrix}, \ k \in \mathbb{N}, \ k \geq 2. \tag{3.4}$$

Според тази дефиниция стълбовете на матрицата G_k са различни, а последният ненулев елемент на някой стълб е 1. Това осигурява стълбовете на G_k да са по двойки непропорционални. Доколкото максимално множество от непропорционални вектори от $\mathbb{F}_q^k$ има $\theta(q,k)$ елемента, колкото е броят на стълбовете на G_k, то G_k е пораждаща матрица на симплекс кода $\mathcal{S}_{q,k}$. Нека стълбовете на матрицата G_k са означени с g_u, $u = 1, \dots, \theta(q,k)$, т. е.

$$G_k = \left(g_1 \dots g_{\theta(q,k)}\right).$$

Разширената матрица $\overline{G}_k$ се дефинира чрез равенството

$$\overline{G}_k = (\mathbf{0}^{\mathrm{T}} | \alpha_1 G_k | \dots | \alpha_{q-1} G_k). \tag{3.5}$$

Тя съдържа като стълбове всички вектори от $\mathbb{F}_q^k$ и *определя наредба* в това множество. Нека стълбовете на матрицата $\overline{G}_k$ са означени с $\bar{g}_t$, $t = 1, \dots, q^k$, т. е.

$$\overline{G}_k = \left(\bar{g}_1 \dots \bar{g}_{q^k}\right).$$

Нека G е пораждаща матрица на линеен $[n,k]_q$ код C с пълна дължина, където $q = 2^m$. Нека f_G е характеристичната функция на C съгласно дефиниция 1.20. *Характеристичният вектор* $\chi = (\chi_1, \dots, \chi_{\theta(q,k)})$ се дефинира чрез равенствата $\chi_u = f_G(g_u)$ за $u = 1, \dots, \theta(q,k)$. *Разширеният характеристичен вектор* $\overline{\chi} = (\overline{\chi}_1, \dots, \overline{\chi}_{q^k})$ се дефинира чрез равенствата $\overline{\chi}_t = f_G(\bar{g}_t)$ за $t = 1, \dots, q^k$. От (3.5) следва, че за всеки $t_1, t_2 \in \mathbb{N}$, за които $1 < t_1 < t_2 \leq q^k$ и $\theta(q,k)$ дели $t_2 - t_1$, векторите $\bar{g}_{t_1}$ и $\bar{g}_{t_2}$ са пропорционални и $\overline{\chi}_{t_1} = f_G(\bar{g}_{t_1}) = f_G(\bar{g}_{t_2}) = \overline{\chi}_{t_2}$. Това показва, че $\overline{\chi} = (0|\chi| \dots |\chi)$.

За обяснение на алгоритъма са нужни следните матрици:

$$\overline{M}_k = \overline{G}_k^{\mathrm{T}} \cdot \overline{G}_k = \left(\langle \bar{g}_{t_1}, \bar{g}_{t_2} \rangle\right)_{t_1,t_2=1}^{q^k},$$

$$M_k = G_k^{\mathrm{T}} \cdot G_k = \left(\langle g_{u_1}, g_{u_2} \rangle\right)_{u_1,u_2=1}^{\theta(q,k)},$$

$$\overline{P}_k = \left((-1)^{\mathrm{Tr}(\langle \bar{g}_{t_1}, \bar{g}_{t_2} \rangle)}\right)_{t_1,t_2=1}^{q^k},$$

$$P_k = \left((-1)^{\mathrm{Tr}(\langle g_{u_1}, g_{u_2}\rangle)}\right)_{u_1,u_2=1}^{\theta(q,k)},$$

$$P_{k,\alpha} = \left((-1)^{\mathrm{Tr}(\alpha\langle g_{u_1}, g_{u_2}\rangle)}\right)_{u_1,u_2=1}^{\theta(q,k)}, \quad \alpha \in \mathbb{F}_q\backslash\{0\},$$

$$\Lambda^{(\alpha)} = \left((-1)^{\mathrm{Tr}(\alpha\alpha_{j_1}\alpha_{j_2})}\right)_{j_1,j_2=0}^{q-1}, \quad \alpha \in \mathbb{F}_q\backslash\{0\}.$$

Матрицата $\overline{P}_k$ може да се получи от матрицата T_k с подходящо разместване на редове и стълбове. Нека $\widehat{\chi} = (\widehat{\chi}_1, \ldots, \widehat{\chi}_{q^k})$ е векторът, определен от равенствата $\widehat{\chi}_t = \widehat{f}(\bar{g}_t)$ за $t = 1, \ldots, q^k$. Тогава $\widehat{\chi}^{\mathrm{T}} = \overline{P}_k \cdot \overline{\chi}^{\mathrm{T}}$.

По (3.5) за матрицата $\overline{M}_k$ се получава

$$\overline{M}_k = \begin{pmatrix} 0 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & \alpha_1^2 M_k & \alpha_1\alpha_2 M_k & \ldots & \alpha_1\alpha_{q-1} M_k \\ \mathbf{0}^{\mathrm{T}} & \alpha_2\alpha_1 M_k & \alpha_2^2 M_k & \ldots & \alpha_2\alpha_{q-1} M_k \\ \vdots & & & & \\ \mathbf{0}^{\mathrm{T}} & \alpha_{q-1}\alpha_1 M_k & \alpha_{q-1}\alpha_2 M_k & \ldots & \alpha_{q-1}^2 M_k \end{pmatrix}, \tag{3.6}$$

откъдето

$$\overline{P}_k = \begin{pmatrix} 1 & \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_1^2} & P_{k,\alpha_1\alpha_2} & \cdots & P_{k,\alpha_1\alpha_{q-1}} \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_2\alpha_1} & P_{k,\alpha_2^2} & \cdots & P_{k,\alpha_2\alpha_{q-1}} \\ \vdots & & & & \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_{q-1}\alpha_1} & P_{k,\alpha_{q-1}\alpha_2} & \cdots & P_{k,\alpha_{q-1}^2} \end{pmatrix}. \tag{3.7}$$

Така

$$\begin{aligned} \widehat{\chi}^{\mathrm{T}} = \overline{P}_k \cdot \overline{\chi}^{\mathrm{T}} &= \begin{pmatrix} 1 & \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_1^2} & P_{k,\alpha_1\alpha_2} & \cdots & P_{k,\alpha_1\alpha_{q-1}} \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_2\alpha_1} & P_{k,\alpha_2^2} & \cdots & P_{k,\alpha_2\alpha_{q-1}} \\ \vdots & & & & \\ \mathbf{1}^{\mathrm{T}} & P_{k,\alpha_{q-1}\alpha_1} & P_{k,\alpha_{q-1}\alpha_2} & \cdots & P_{k,\alpha_{q-1}^2} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \chi^{\mathrm{T}} \\ \chi^{\mathrm{T}} \\ \vdots \\ \chi^{\mathrm{T}} \end{pmatrix} \\ &= \begin{pmatrix} (q-1)\sum_{u=1}^{\theta(q,k)} \chi_u \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_1\alpha_j}\right)\chi^{\mathrm{T}} \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_2\alpha_j}\right)\chi^{\mathrm{T}} \\ \vdots \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_{q-1}\alpha_j}\right)\chi^{\mathrm{T}} \end{pmatrix}. \end{aligned} \tag{3.8}$$

Като се има предвид, че умножението по елемент на полето индуцира пермутация на елементите на полето със запазване на нулата, то

$$\overline{P}_k \cdot \overline{\chi}^{\mathrm{T}} = \begin{pmatrix} (q-1)\sum_{u=1}^{\theta(q,k)} \chi_u \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \\ \vdots \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \end{pmatrix}. \tag{3.9}$$

Това означава, че не е нужно да се използват матрицата $\overline{P}_k$ с размери $2^k \times 2^k$ и дългият вектор $\overline{\chi}$. За получаване на $\widehat{\chi}$ и тегловното разпределение на кода е достатъчно да се използва характеристичният вектор χ, матрицата P_k и матриците $P_{k,\alpha}$ за $\alpha \in \mathbb{F}_q \backslash \{0\}$.

От (3.4) за матрицата $M_k = G_k^{\mathrm{T}} \cdot G_k$ се получава следната рекурентна връзка:

$$M_k = \begin{pmatrix} M_{k-1} & M_{k-1} & \cdots & M_{k-1} & \mathbf{0}^{\mathrm{T}} \\ M_{k-1} & \alpha_1^2 J + M_{k-1} & \cdots & \alpha_1\alpha_{q-1} J + M_{k-1} & \boldsymbol{\alpha_1}^{\mathrm{T}} \\ M_{k-1} & \alpha_2\alpha_1 J + M_{k-1} & \ldots & \alpha_2\alpha_{q-1} J + M_{k-1} & \boldsymbol{\alpha_2}^{\mathrm{T}} \\ \vdots & & & & \\ M_{k-1} & \alpha_{q-1}\alpha_1 J + M_{k-1} & \cdots & \alpha_{q-1}^2 J + M_{k-1} & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}} \\ \mathbf{0} & \boldsymbol{\alpha_1} & \ldots & \boldsymbol{\alpha_{q-1}} & 1 \end{pmatrix}, \tag{3.10}$$

където J е матрица, съдържаща само единици, от съответната размерност.

Като се ползва рекурентната връзка (3.10), се получава

$$P_k = \begin{pmatrix} P_{k-1} & P_{k-1} & \cdots & P_{k-1} & \boldsymbol{\Lambda_0}^{\mathrm{T}} \\ P_{k-1} & \Lambda_{\alpha_1^2} P_{k-1} & \cdots & \Lambda_{\alpha_1\alpha_{q-1}} P_{k-1} & \boldsymbol{\Lambda_{\alpha_1}}^{\mathrm{T}} \\ P_{k-1} & \Lambda_{\alpha_2\alpha_1} P_{k-1} & \cdots & \Lambda_{\alpha_2\alpha_{q-1}} P_{k-1} & \boldsymbol{\Lambda_{\alpha_2}}^{\mathrm{T}} \\ \vdots & & & & \\ P_{k-1} & \Lambda_{\alpha_{q-1}\alpha_1} P_{k-1} & \cdots & \Lambda_{\alpha_{q-1}^2} P_{k-1} & \boldsymbol{\Lambda_{\alpha_{q-1}}}^{\mathrm{T}} \\ \boldsymbol{\Lambda_0} & \boldsymbol{\Lambda_{\alpha_1}} & \cdots & \boldsymbol{\Lambda_{\alpha_{q-1}}} & \Lambda_1 \end{pmatrix}, \tag{3.11}$$

$$P_k = \begin{pmatrix} & & & \boldsymbol{\Lambda_0}^{\mathrm{T}} \\ & T_1 \otimes P_{k-1} & & \boldsymbol{\Lambda_{\alpha_1}}^{\mathrm{T}} \\ & & & \vdots \\ & & & \boldsymbol{\Lambda_{\alpha_{q-1}}}^{\mathrm{T}} \\ \boldsymbol{\Lambda_0} & \boldsymbol{\Lambda_{\alpha_1}} & \ldots \quad \boldsymbol{\Lambda_{\alpha_{q-1}}} & \Lambda_1 \end{pmatrix}, \tag{3.12}$$

където $\Lambda_\alpha = (-1)^{\mathrm{Tr}(\alpha)}$ за $\alpha \in \mathbb{F}_q$, а $\boldsymbol{\Lambda_\alpha}$ е вектор ред с дължина по подразбиране и едни и същи координати Λ_α. Последното равенство и (3.2) дават възможност за прилагане на бътерфлай алгоритъм за пресмятането на $P_k \cdot \chi^{\mathrm{T}}$.

Нека характеристичният вектор χ е разделен на части, както следва

$$\chi = (\chi^{(0)}|\chi^{(1)}|\ldots|\chi^{(q-1)}|\chi_{\theta(q,k)}), \tag{3.13}$$

където $\chi^{(0)}, \chi^{(1)}, \ldots, \chi^{(q-1)} \in \mathbb{Z}^{\theta(q,k-1)}$. Изпълнено е

$$\begin{aligned} P_k \cdot \chi^{\mathrm{T}} &= \begin{pmatrix} (T_1 \otimes P_{k-1}) \cdot \begin{pmatrix} {\chi^{(0)}}^{\mathrm{T}} \\ {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} + \chi_\theta \begin{pmatrix} {\boldsymbol{\Lambda_0}}^{\mathrm{T}} \\ {\boldsymbol{\Lambda_{\alpha_1}}}^{\mathrm{T}} \\ \vdots \\ {\boldsymbol{\Lambda_{\alpha_{q-1}}}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \Lambda_{\alpha_1} \sum \chi^{(1)} + \cdots + \Lambda_{\alpha_{q-1}} \sum \chi^{(q-1)} + \Lambda_1 \chi_\theta \end{pmatrix} \\ &= \begin{pmatrix} (T_1 \otimes I_{\theta(q,k-1)}) \cdot \begin{pmatrix} P_{k-1} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{k-1} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{k-1} \cdot {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} + \chi_\theta \begin{pmatrix} {\boldsymbol{\Lambda_0}}^{\mathrm{T}} \\ {\boldsymbol{\Lambda_{\alpha_1}}}^{\mathrm{T}} \\ \vdots \\ {\boldsymbol{\Lambda_{\alpha_{q-1}}}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \Lambda_{\alpha_1} \sum \chi^{(1)} + \cdots + \Lambda_{\alpha_{q-1}} \sum \chi^{(q-1)} + \Lambda_1 \chi_\theta \end{pmatrix}, \end{aligned} \tag{3.14}$$

където за краткост $\theta = \theta(q,k)$ и $\sum \chi^{(j)}$ означава сумата от координатите на $\chi^{(j)}$, $j = 0, 1, \ldots, q-1$.

Нека $Num(\alpha) \in \{0, 1, \ldots, q-1\}$ е номерът на елемента α в наредбата на полето $\mathbb{F}_q$, т. е. $\alpha_{Num(\alpha)} = \alpha$. В равенство (3.14) може да се забележи, че събираемите с участието на χ_θ са подложени на същите операции, както събираемите от групата на $\chi^{(Num(1))}$. Това е така, понеже

$$\chi_\theta \begin{pmatrix} {\boldsymbol{\Lambda_0}}^{\mathrm{T}} \\ {\boldsymbol{\Lambda_{\alpha_1}}}^{\mathrm{T}} \\ \vdots \\ {\boldsymbol{\Lambda_{\alpha_{q-1}}}}^{\mathrm{T}} \end{pmatrix} = \chi_\theta \begin{pmatrix} \Lambda_0 \\ \Lambda_{\alpha_1} \\ \vdots \\ \Lambda_{\alpha_{q-1}} \end{pmatrix} \otimes \mathbf{1}^{\mathrm{T}} = \left(\begin{pmatrix} \Lambda_0 \\ \Lambda_{\alpha_1} \\ \vdots \\ \Lambda_{\alpha_{q-1}} \end{pmatrix} \otimes I_{\theta(q,k-1)} \right) \cdot (\chi_\theta . \mathbf{1}^{\mathrm{T}}),$$

а $(\Lambda_0, \Lambda_{\alpha_1}, \ldots, \Lambda_{\alpha_{q-1}})^{\mathrm{T}}$ е стълбът, съответстващ на $\alpha = 1$ в матрицата T_1.

Оттук

$$P_k \cdot \chi^{\mathrm{T}} = \begin{pmatrix} (T_1 \otimes I_{\theta(q,k-1)}) \cdot \begin{pmatrix} P_{k-1} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{k-1} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{k-1} \cdot {\chi^{(Num(1))}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1} \cdot {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \cdots + \Lambda_1 \left(\chi_\theta + \sum \chi^{(Num(1))} \right) + \cdots + \Lambda_{\alpha_{q-1}} \sum \chi^{(q-1)} \end{pmatrix}. \quad (3.15)$$

За $1 < l < k$ е изпълнено

$$P_l \cdot {\chi'}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} = \begin{pmatrix} \left(T_1 \otimes I_{\theta(q,l-1)}\right) \cdot \begin{pmatrix} P_{l-1} \cdot {\chi'^{(0)}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ P_{l-1} \cdot {\chi'^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{l-1} \cdot {\chi'^{(Num(1))}}^{\mathrm{T}} + \chi'_{\theta(q,l-1)}.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{l-1} \cdot {\chi'^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \left(\chi_\theta + \sum \chi'^{(0)}\right) + \Lambda_{\alpha_1} \sum \chi'^{(1)} + \ldots \end{pmatrix}, \quad (3.16)$$

където χ' е вектор с дължина $\theta(q,l)$, който е подходяща част от χ. От това равенство по индукция се доказва, че за получаване на $P_{k-1} \cdot {\chi^{(Num(1))}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}}$ е достатъчно да се добави χ_θ само към първата координата на вектора $\chi^{(Num(1))}$, но предварително умножено по $\Lambda_1 = \Lambda_1^{-1}$. Особеност в базата на индукцията е, че $P_1 = (\Lambda_1)$ и

$$P_1 \cdot \chi_1^{(Num(1))} + \chi_\theta = \Lambda_1 (\chi_1^{(Num(1))} + \Lambda_1^{-1} \chi_\theta).$$

За пресмятане на последната координата на $P_k \cdot \chi^{\mathrm{T}}$ е необходимо да се добави χ_θ към същата координата, но без предварително умножение. Тези наблюдения дават възможност да се извършат предварителни операции със стойността на χ_θ, както и с последните координати от всеки блок $\chi'^{(j)}$, $j = 0, \ldots, q-1$.

Пример 3.2. Нека $q = 4$, $\mathbb{F}_4 = \{0, \alpha_1 = \beta_2, \alpha_2 = \beta_1, \alpha_3 = 1\}$, $k = 3$ и $\chi = (4,5,2,8,9,3,7,4,5,3,3,5,4,7,4,5,5,8,4,3,1)$. Това е характеристичен вектор на линеен код над $\mathbb{F}_4$ с дължина $n = 99$ и размерност 3. Тогава

$$T_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

$$P_3 \cdot \chi^{\mathrm{T}} = \begin{pmatrix} (T_1 \otimes I_5) \cdot \begin{pmatrix} P_2 \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_2 \cdot {\chi^{(1)}}^{\mathrm{T}} \\ P_2 \cdot {\chi^{(2)}}^{\mathrm{T}} \\ P_2 \cdot {\chi^{(3)}}^{\mathrm{T}} + \mathbf{1}^{\mathrm{T}} \end{pmatrix} \\ \sum \chi^{(0)} - \sum \chi^{(1)} - \sum \chi^{(2)} + \sum \chi^{(3)} + 1 \end{pmatrix}.$$

Тъй като $T_1 \otimes I_1 = T_1$ и $P_1 = (1)$, изпълнено е

$$P_2 \cdot {\chi^{(j)}}^{\mathrm{T}} = \begin{pmatrix} T_1 \cdot (\chi_1^{(j)}, \chi_2^{(j)}, \chi_3^{(j)}, \chi_4^{(j)} + \chi_5^{(j)})^{\mathrm{T}} \\ \chi_1^{(j)} - \chi_2^{(j)} - \chi_3^{(j)} + \chi_4^{(j)} + \chi_5^{(j)} \end{pmatrix}, \quad j = 0, 1, 2,$$

$$P_2 \cdot {\chi^{(3)}}^{\mathrm{T}} + \mathbf{1}^{\mathrm{T}} = \begin{pmatrix} T_1 \cdot (\chi_1^{(3)} + 1, \chi_2^{(3)}, \chi_3^{(3)}, \chi_4^{(3)} + \chi_5^{(3)})^{\mathrm{T}} \\ \chi_1^{(3)} - \chi_2^{(3)} - \chi_3^{(3)} + \chi_4^{(3)} + \chi_5^{(3)} + 1 \end{pmatrix}.$$

След пресмятането се получава $(P_3 \cdot \chi^{\mathrm{T}})^{\mathrm{T}} = \chi \cdot P_3^{\mathrm{T}} =$
$(99, -31, -23, 19, 19, 3, -33, -11, 19, 19, 1, -17, -1, 9, 9, 9, 3, \ -5, 9, 9, 9)$.

Следва да се обърне внимание на матриците $P_{k,\alpha}$ за $\alpha \in \mathbb{F}_q \backslash \{0\}$. За $k = 1$ матриците са $P_{1,\alpha} = (\Lambda_\alpha)$. При рекурентната стъпка е в сила

$$P_{k,\alpha} = \begin{pmatrix} & & \boldsymbol{\Lambda_0}^{\mathrm{T}} \\ \Lambda^{(\alpha)} \otimes P_{k-1,\alpha} & & \boldsymbol{\Lambda_{\alpha\alpha_1}}^{\mathrm{T}} \\ & & \vdots \\ & & \boldsymbol{\Lambda_{\alpha\alpha_{q-1}}}^{\mathrm{T}} \\ \boldsymbol{\Lambda_0} \quad \boldsymbol{\Lambda_{\alpha\alpha_1}} & \cdots \quad \boldsymbol{\Lambda_{\alpha\alpha_{q-1}}} & \Lambda_\alpha \end{pmatrix}. \tag{3.17}$$

Тогава

$$P_{k,\alpha} \cdot \chi^{\mathrm{T}} = \begin{pmatrix} (\Lambda^{(\alpha)} \otimes I_{\theta(q,k-1)}) \cdot \begin{pmatrix} P_{k-1,\alpha} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{k-1,\alpha} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha} \cdot {\chi^{(Num(1))}}^{\mathrm{T}} + \chi_\theta . \mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha} \cdot {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \Lambda_{\alpha\alpha_1} \sum \chi^{(1)} + \cdots + \Lambda_\alpha \left(\chi_\theta + \sum \chi^{(Num(1))} \right) + \cdots \end{pmatrix}. \tag{3.18}$$

Тъй като умножението по $\alpha \neq 0$ може да се разгледа като пермутация на елементите на $\mathbb{F}_q$, матрицата $\Lambda^{(\alpha)}$ може да бъде получена от матрицата $T_1 = \Lambda^{(1)}$ чрез подходяща пермутация на редове (и/или стълбове). Нека $\pi_\alpha \in S_q$

е пермутация, дефинирана с равенството $\pi_\alpha(j) = j'$, където $\alpha_{j'} = \alpha\alpha_j$, $j = 0, 1, \ldots, q-1$. Пермутацията π_α индуцира пермутация на блоковете от стълбове в матрицата $T_1 \otimes I_{\theta(q,k-1)}$. Затова

$$P_{k,\alpha} \cdot \chi^{\mathrm{T}} = \begin{pmatrix} (T_1 \otimes I_{\theta(q,k-1)}) \cdot \begin{pmatrix} P_{k-1,\alpha} \cdot \chi^{(\pi_\alpha^{-1}(0))^{\mathrm{T}}} \\ P_{k-1,\alpha} \cdot \chi^{(\pi_\alpha^{-1}(1))^{\mathrm{T}}} \\ \vdots \\ P_{k-1,\alpha} \cdot \chi^{(\pi_\alpha^{-1}(Num(\alpha)))^{\mathrm{T}}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha} \cdot \chi^{(\pi_\alpha^{-1}(q-1))^{\mathrm{T}}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \Lambda_{\alpha\alpha_1} \sum \chi^{(1)} + \cdots + \Lambda_\alpha \left(\chi_\theta + \sum \chi^{(Num(1))}\right) + \cdots \end{pmatrix}, \tag{3.19}$$

като $\pi_\alpha^{-1}(Num(\alpha)) = Num(1)$.

С аналогични разсъждения, както в случая $\alpha = 1$, може да се заключи, че за добавяне на χ_θ към всички координати на $P_{k-1,\alpha} \cdot \chi^{(Num(1))^{\mathrm{T}}}$ е достатъчно да се добави χ_θ, предварително умножено по Λ_α, към първата координата на вектора $\chi^{(Num(1))}$. За пресмятане на последната координата на $P_{k,\alpha} \cdot \chi^{\mathrm{T}}$ е необходимо да се добави χ_θ към първата координата на вектора $\chi^{(Num(1))}$, но без предварително умножение. Подобно се постъпва и с последните координати на междинните блокове. Специфично в обосновката е, че $\pi_\alpha(0) = 0$.

Сравнението на равенства (3.15) и (3.19) показва, че за пресмятане на координатите без последната се ползва умножение по една и съща матрица $T_1 \otimes I_{\theta(q,k-1)}$. В последното равенство блоковете $P_{k-1,\alpha} \cdot \chi^{(j)^{\mathrm{T}}}$ са подложени на пермутация. По индукция, тази пермутация може да се пренесе върху координатите на вектора χ, което е описано подробно в раздел 3.4.1. Също така, χ_θ и последните елементи на междинните блокове се добавят на определени места в χ. Така модифицираният вектор по-долу е означен с $\pi_\alpha(\chi)$.

В разработения подобрен алгоритъм, за да се пресметне $\left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}}$, директно се манипулира със сумата

$$SP(\chi) = \pi_{\alpha_1}(\chi) + \pi_{\alpha_2}(\chi) + \cdots + \pi_{\alpha_{q-1}}(\chi),$$

като се ползва допълнително само една модификация на χ.

Пример 3.3. Кодът е от пример 3.2, но се очаква да се пресметне $P_{3,\alpha_1} \cdot \chi^T$.

Тъй като $\pi_{\alpha_1} = (1\ 2\ 3)$, то

$$P_{3,\alpha_1} \cdot \chi^T = \begin{pmatrix} (\Lambda^{(\alpha_1)} \otimes I_5) \begin{pmatrix} P_{2,\alpha_1} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(2)}}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(3)}}^{\mathrm{T}} + \mathbf{1}^{\mathrm{T}} \end{pmatrix} \\ \sum \chi^{(0)} - \sum \chi^{(1)} + \sum \chi^{(2)} - \left(\sum \chi^{(3)} + 1\right) \end{pmatrix}$$

$$= \begin{pmatrix} (T_1 \otimes I_5) \begin{pmatrix} P_{2,\alpha_1} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(3)}}^{\mathrm{T}} + \mathbf{1}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ P_{2,\alpha_1} \cdot {\chi^{(2)}}^{\mathrm{T}} \end{pmatrix} \\ \sum \chi^{(0)} - \sum \chi^{(1)} + \sum \chi^{(2)} - \left(\sum \chi^{(3)} + 1\right) \end{pmatrix}.$$

Освен това, $\Lambda^{(\alpha_1)} \otimes I_1 = \Lambda^{(\alpha_1)}$ и $P_{1,\alpha_1} = (-1)$, откъдето

$$P_{2,\alpha_1} \cdot {\chi^{(j)}}^{\mathrm{T}} = \begin{pmatrix} \Lambda^{(\alpha_1)} \cdot \left(-\chi_1^{(j)}, -\chi_2^{(j)}, -\chi_3^{(j)}, -\left(\chi_4^{(j)} - \chi_5^{(j)}\right)\right)^{\mathrm{T}} \\ \chi_1^{(j)} - \chi_2^{(j)} + \chi_3^{(j)} - \left(\chi_4^{(j)} + \chi_5^{(j)}\right) \end{pmatrix}$$

$$= \begin{pmatrix} T_1 \cdot (-\chi_1^{(j)}, -\chi_4^{(j)} + \chi_5^{(j)}, -\chi_2^{(j)}, -\chi_3^{(j)})^{\mathrm{T}} \\ \chi_1^{(j)} - \chi_2^{(j)} + \chi_3^{(j)} - \chi_4^{(j)} - \chi_5^{(j)} \end{pmatrix},\ j = 0, 1, 2,$$

$$P_{2,\alpha_1} \cdot {\chi^{(3)}}^{\mathrm{T}} + \mathbf{1}^{\mathrm{T}} = \begin{pmatrix} {}^{\mathrm{T}}\Lambda^{(\alpha_1)} \cdot \left(-\chi_1^{(3)} + 1, -\chi_2^{(3)}, -\chi_3^{(3)}, -\left(\chi_4^{(3)} - \chi_5^{(3)}\right)\right)^{\mathrm{T}} \\ \left(\chi_1^{(3)} + 1\right) - \chi_2^{(3)} + \chi_3^{(3)} - \left(\chi_4^{(3)} + \chi_5^{(3)}\right) \end{pmatrix}$$

$$= \begin{pmatrix} T_1 \cdot (-\chi_1^{(3)} + 1, -\chi_4^{(3)} + \chi_5^{(3)}, -\chi_2^{(3)}, -\chi_3^{(3)})^{\mathrm{T}} \\ \chi_1^{(3)} + 1 - \chi_2^{(3)} + \chi_3^{(3)} - \chi_4^{(3)} - \chi_5^{(3)} \end{pmatrix}.$$

Следователно $(P_{3,\alpha_1} \cdot \chi^{\mathrm{T}})^{\mathrm{T}} =$
$(-59, -13, 21, -5, -31, 9, -5, -7, 3, -19, 7, -11, -1, 5, -17, 3, -3, 3, -11, 3, 3)$.

3.4 Описание на подобрения алгоритъм

3.4.1 Предварителни изчисления

Нека $\rho : \mathbb{Z}_{\theta(q,k)} \to \mathbb{Z}^{k-1}$ е изображение, което се дефинира, както следва: ако $0 \leq z \leq \theta(q,k) - 1$ и $\rho(z) = (\rho_1, \ldots, \rho_{k-1})$, то

$$\rho_1 = \left\lfloor \frac{z}{\theta(q,k-1)} \right\rfloor, \quad \rho_t = \left\lfloor \frac{z - \sum_{s=1}^{t-1} \rho_s \theta(q,k-s)}{\theta(q,k-t)} \right\rfloor \text{ за } t = 2, \ldots, k-1.$$

Ще бъдат описани свойствата на това изображение.

Лема 3.5. *Ако* $0 \leq z \leq \theta(q,k) - 1$ *и* $\rho(z) = (\rho_1, \ldots, \rho_{k-1})$, *то*

$$z = \rho_1\theta(q,k-1) + \rho_2\theta(q,k-2) + \cdots + \rho_{k-2}\theta(q,2) + \rho_{k-1},$$

и $0 \leq \rho_t \leq q$, $t = 1, \ldots, k-1$.

Доказателство. Нека $r_t = z - \sum_{s=1}^{t} \rho_s \theta(q,k-s)$ за $t = 1, 2, \ldots, k-2$, т. е.

$$\rho_t = \left\lfloor \frac{r_{t-1}}{\theta(q,k-t)} \right\rfloor \text{ за } t = 2, \ldots, k-1.$$

С индукция по t ще бъде показано, че $0 \leq r_t < \theta(q,k-t)$ и $0 \leq \rho_t \leq q$ за $t = 1, 2, \ldots, k-1$.

От дефиницията $z = \rho_1\theta(q,k-1) + r_1$, където r_1 е остатъкът от целочисленото деление, за който $0 \leq r_1 < \theta(q,k-1)$. Действително,

$$\rho_1 \leq \frac{z}{\theta(q,k-1)} < \rho_1 + 1$$

$$\theta(q,k-1)\rho_1 \leq z < \theta(q,k-1)\rho_1 + \theta(q,k-1)$$

$$0 \leq r_1 = z - \theta(q,k-1)\rho_1 < \theta(q,k-1).$$

От $z \leq \theta(q,k) - 1$ и $\theta(q,k) = q\theta(q,k-1) + 1$ се получава

$$\rho_1 = \left\lfloor \frac{z}{\theta(q,k-1)} \right\rfloor \leq \frac{q\theta(q,k-1)}{\theta(q,k-1)} = q.$$

От друга страна, $z \geq 0$ и следователно $\rho_1 \geq 0$.

Нека $t \in \mathbb{Z}$, $1 \leq t < k-1$, е фиксирано, като за него е изпълнено $0 \leq r_t < \theta(q,k-t)$ и $0 \leq \rho_t \leq q$. Същото твърдение е в сила за $t+1$. Наистина, $r_{t+1} =$

$z-\sum_{s=1}^{t+1}\rho_s\theta(q,k-s)=r_t-\rho_{t+1}\theta(q,k-t-1)$, откъдето $r_t=\rho_{t+1}\theta(q,k-t-1)+r_{t+1}$. От друга страна

$$\rho_{t+1}\le\frac{r_t}{\theta(q,k-t-1)}<\rho_{t+1}+1$$

$$\theta(q,k-t-1)\rho_{t+1}\le r_t<\theta(q,k-t-1)\rho_{t+1}+\theta(q,k-t-1)$$

$$0\le r_{t+1}=r_t-\theta(q,k-t-1)\rho_t<\theta(q,k-t-1).$$

От $r_t<\theta(q,k-t)=q\theta(q,k-t-1)+1$ следва

$$\rho_{t+1}=\left\lfloor\frac{r_t}{\theta(q,k-t-1)}\right\rfloor\le\frac{q\theta(q,k-t-1)}{\theta(q,k-t-1)}=q,$$

а от $r_t\ge 0$ се извежда $\rho_{t+1}\ge 0$.

Тъй като $\theta(q,1)=1$, за ρ_{k-1} е изпълнено

$$\rho_{k-1}=\left\lfloor\frac{r_{k-2}}{\theta(q,1)}\right\rfloor=r_{k-2}.$$

Окончателно

$$\begin{aligned}z &= \rho_1\theta(q,k-1)+r_1=\rho_1\theta(q,k-1)+\rho_2\theta(q,k-2)+r_2\\ &= \cdots=\rho_1\theta(q,k-1)+\rho_2\theta(q,k-2)+\cdots+\rho_{k-2}\theta(q,2)+r_{k-2}\\ &= \rho_1\theta(q,k-1)+\rho_2\theta(q,k-2)+\cdots+\rho_{k-2}\theta(q,2)+\rho_{k-1}.\end{aligned}$$

□

Следствие 3.6. *Изображението ρ е инективно.*

Лема 3.7. *Ако $0\le z\le\theta(q,k)-1$, $\rho(z)=(\rho_1,\ldots,\rho_{k-1})$ и съществува индекс $t\in\{1,\ldots,k-1\}$, така че $\rho_t=q$ и $\rho_s<q$ за $s=1,\ldots,t-1$, то $\rho_{t+1}=\cdots=\rho_{k-1}=0$.*

Доказателство. Нека $r_0=z$, $r_{s-1}=\rho_s\theta(q,k-s)+r_s$, $r_s<\theta(q,k-s)$, $s=1,\ldots,k-1$. Ако $\rho_t=q$, то

$$r_{t-1}=q\theta(q,k-t)+r_t<\theta(q,k-t+1)=q\theta(q,k-t)+1.$$

Следователно $r_t<1$, а оттук $r_t=0$. Това води до $r_{t+1}=\cdots=r_{k-1}=0$ и $\rho_{t+1}=\cdots=\rho_{k-1}=0$. □

Малка модификация на вектора $\rho(z)$ е по-удобна за разработения алгоритъм. За целта се ползва изображението $\nu : \{1, \ldots, \theta(q,k)\} \to \mathbb{Z}^{k-1}$, дефинирано чрез

$$\nu(z) = \begin{cases} \rho(z), & \text{ако } \rho_t < q \text{ за всяко } t = 1, \ldots, k-1, \\ (\rho_1, \ldots, \rho_{t-1}, q, \ldots, q), & \text{ако } \rho_t = q \text{ за някое } t \leq k-1. \end{cases}$$

Нека $\kappa : \mathbb{F}_q^{k-1} \to \mathbb{Z}$ е изображение, дефинирано с равенството

$$\kappa(\alpha_{j_1}, \ldots, \alpha_{j_{k-1}}) = \rho^{-1}(j_1, \ldots, j_{k-1}) + 1.$$

Очевидно, образите на векторите от $\mathbb{F}_q^{k-1}$ са естествени числа, по-малки или равни на $\theta(q,k)$. При това, различните вектори имат различни образи. Ако u, за което $1 \leq u \leq \theta(q,k)$, не е образ при изображението κ, съответната координата χ_u на характеристичния вектор χ се нарича *неактивна координата*. Всъщност, χ_u е неактивна координата, ако последната координата на вектора $\nu(u-1)$ е q.

Изображенията $\rho^{(l)} : \mathbb{Z}_{\theta(q,l)} \to \mathbb{Z}^{l-1}$, $\nu^{(l)}$ и $\kappa^{(l)} : \mathbb{F}_q^{l-1} \to \mathbb{Z}$ се дефинират по същия начин, както ρ, ν и κ съответно, но с l вместо k, за $l = 2, \ldots, k$.

Нека $2 \leq l \leq k$ и с $b(l,0), \ldots, b(l, q^{l-1}-1)$ са означени векторите от $\mathbb{F}_q^{l-1}$, наредени лексикографски. Нека още $\sigma_l : \mathbb{F}_q^{l-1} \to \mathbb{Z}^{l-1}$ е изображение, за което

$$\sigma_l(\alpha_{j_1}, \ldots, \alpha_{j_{l-1}}) = (j_1, \ldots, j_{l-1}).$$

В разработеният алгоритъм се ползват три масива с дължина $\theta(q,k)$, означени с $\chi(0)$, $\chi(1)$ и S, чието формиране е описано в Алгоритъм 5. Масивите $\chi(0)$ и $\chi(1)$ играят роля на модифицирани копия на характеристичния вектор χ. Ако последната координата на $\nu(u-1)$ не е q, то $\chi(s)[u] = (-1)^s \chi_u$, за $u = 1, \ldots, \theta(q,k)$ и $s = 0, 1$. Неактивните координати на χ се добавят на подходящи места в копията, като процесът е описан на редове 4–9 в Алгоритъм 5. Масивът S служи за формиране на вектора $SP(\chi)$.

Пример 3.4. За кода от пример 3.2 с характеристичен вектор

$$\chi = (4,5,2,8,9,3,7,4,5,3,3,5,4,7,4,5,5,8,4,3,1)$$

масивите, получени по Алгоритъм 5, са

$$\begin{aligned} \chi(0) &= (4,5,2,17,0,3,7,4,8,0,3,5,4,11,0,6,5,8,7,0,0); \\ \chi(1) &= (-4,-5,-2,1,0,-3,-7,-4,-2,0,-3,-5,-4,-3,0,-4,-5,-8,-1,0,0); \\ S &= (-4,4,-2,10,0,-4,2,-4,-5,0,-4,-5,-4,2,0,0,-2,1,-4,0,0). \end{aligned}$$

Стойностите в неактивните координати на $\chi(0)$, $\chi(1)$ и S са нули. В схемите по-долу е показано приложението на Алгоритъм 5.

Алгоритъм 5 Предварителни изчисления

Вход: естествени числа q, k и масив χ с дължина $\theta(q,k)$

Изход: масиви $\chi(0)$ и S с дължина $\theta(q,k)$

1: Инициализиране с 0 на масиви $\chi(0), \chi(1), S$ с дължина $\theta(q,k)$;
2: **for** $u = 1$ **to** $\theta(q,k)$ **do**
3: $b' = \nu(u-1)$; $t = k-1$;
4: **while** $b'[t] = q$ and $t > 0$ **do**
5: $b'[t] = 0$; $t = t-1$;
6: **end while**
7: **if** $t < k-1$ **then**
8: $b'[t+1] = Num(1)$; $u_1 = \nu^{-1}(b') + 1$;
9: $\chi(0)[u_1] = \chi(0)[u_1] + \chi[u]$; $\chi(1)[u_1] = \chi(1)[u_1] + \chi[u]$;
10: **else**
11: $\chi(0)[u] = \chi[u]$; $\chi(1)[u] = -\chi[u]$;
12: **end if**
13: **end for**
14: **for** $j = 1$ **to** $q-1$ **do**
15: $a = \mathrm{Tr}(\alpha_j)$;
16: **for** $i = 0$ **to** $q^{k-1}-1$ **do**
17: $u = \kappa(b(k,i))$; $u_1 = \kappa(\alpha_j b(k,i))$; $S[u_1] = S[u_1] + \chi(a)[u]$;
18: **end for**
19: **end for**

u	$\nu(u-1)$	b'	χ	$\chi(0)$	$\chi(1)$	$\chi(0)$	$\chi(1)$
1	00		4	4	−4	4	−4
2	01		5	5	−5	5	−5
3	02		2	2	−2	2	−2
4	03		8	8 +9	−8 +9	17	1
5	04	03	9				
6	10		3	3	−3	3	−3
7	11		7	7	−7	7	−7
8	12		4	4	−4	4	−4
9	13		5	5 +3	−5 +3	8	−2
10	14	13	3				
11	20		3	3	−3	3	−3
12	21		5	5	−5	5	−5
13	22		4	4	−4	4	−4
14	23		7	7 +4	−7 +4	11	−3
15	24	23	4				
16	30		5	5 +1	−5 +1	6	−4
17	31		5	5	−5	5	−5
18	32		8	8	−8	8	−8
19	33		4	4 +3	−4 +3	7	−1
20	34	33	3				
21	44	30	1				

Multiplication in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_0	α_0	α_0	α_0	α_0
α_1	α_0	α_2	α_3	α_1
α_2	α_0	α_3	α_1	α_2
α_3	α_0	α_1	α_2	α_3

Inverse in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_j^{-1}	$-$	α_2	α_1	α_3

Traces in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
Tr	0	1	1	0

$\chi(0)$	$\chi(1)$	$\sigma_k(b)$	$\sigma_k(\alpha_1 b)$	$\sigma_k(\alpha_2 b)$	$\pi_{\alpha_1}(\chi)$
4	−4	00	00	00	−4
5	−5	01	02	03	1
2	−2	02	03	01	−5
17	1	03	01	02	−2
3	−3	10	20	30	−4
7	−7	11	22	33	−1
4	−4	12	23	31	−5
8	−2	13	21	32	−8
3	−3	20	30	10	−3
5	−5	21	32	13	−2
4	−4	22	33	11	−7
11	−3	23	31	12	−4
6	−4	30	10	20	−3
5	−5	31	12	23	−3
8	−8	32	13	21	−5
7	−1	33	11	22	−4

Multiplication in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_0	α_0	α_0	α_0	α_0
α_1	α_0	α_2	α_3	α_1
α_2	α_0	α_3	α_1	α_2
α_3	α_0	α_1	α_2	α_3

Inverse in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_j^{-1}	$-$	α_2	α_1	α_3

Traces in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
Tr	0	1	1	0

$\chi(0)$	$\chi(1)$	$\sigma_k(b)$	$\pi_{\alpha_1}(\chi)$	$\sigma_k(\alpha_1 b)$	$\pi_{\alpha_2}(\chi)$
4	−4	00	−4	00	−4
5	−5	01	1	02	−2
2	−2	02	−5	03	1
17	1	03	−2	01	−5
3	−3	10	−4	20	−3
7	−7	11	−1	22	−4
4	−4	12	−5	23	−3
8	−2	13	−8	21	−5
3	−3	20	−3	30	−4
5	−5	21	−2	32	−8
4	−4	22	−7	33	−1
11	−3	23	−4	31	−5
6	−4	30	−3	10	−3
5	−5	31	−3	12	−4
8	−8	32	−5	13	−2
7	−1	33	−4	11	−7

Multiplication in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_0	α_0	α_0	α_0	α_0
α_1	α_0	α_2	α_3	α_1
α_2	α_0	α_3	α_1	α_2
α_3	α_0	α_1	α_2	α_3

Inverse in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_j^{-1}	$-$	α_2	α_1	α_3

Traces in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
Tr	0	1	1	0

$\chi(0)$	$\chi(1)$	$\sigma_k(b)$	$\pi_{\alpha_1}(\chi)$	$\pi_{\alpha_2}(\chi)$	$\sigma_k(\alpha_3 b)$	$\pi_{\alpha_3}(\chi)$
4	−4	00	−4	−4	00	4
5	−5	01	1	−2	01	5
2	−2	02	−5	1	02	2
17	1	03	−2	−5	03	17
3	−3	10	−4	−3	10	3
7	−7	11	−1	−4	11	7
4	−4	12	−5	−3	12	4
8	−2	13	−8	−5	13	8
3	−3	20	−3	−4	20	3
5	−5	21	−2	−8	21	5
4	−4	22	−7	−1	22	4
11	−3	23	−4	−5	23	11
6	−4	30	−3	−3	30	6
5	−5	31	−3	−4	31	5
8	−8	32	−5	−2	32	8
7	−1	33	−4	−7	33	7

Multiplication in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_0	α_0	α_0	α_0	α_0
α_1	α_0	α_2	α_3	α_1
α_2	α_0	α_3	α_1	α_2
α_3	α_0	α_1	α_2	α_3

Inverse in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
α_j^{-1}	$-$	α_2	α_1	α_3

Traces in $\mathbb{F}_q$

	α_0	α_1	α_2	α_3
Tr	0	1	1	0

$\chi(0)$	$\chi(1)$	$\sigma_k(b)$	$\pi_{\alpha_1}(\chi)$	$\pi_{\alpha_2}(\chi)$	$\pi_{\alpha_3}(\chi)$	S
4	−4	00	−4	−4	4	−4
5	−5	01	1	−2	5	4
2	−2	02	−5	1	2	−2
17	1	03	−2	−5	17	10
3	−3	10	−4	−3	3	−4
7	−7	11	−1	−4	7	2
4	−4	12	−5	−3	4	−4
8	−2	13	−8	−5	8	−5
3	−3	20	−3	−4	3	−4
5	−5	21	−2	−8	5	−5
4	−4	22	−7	−1	4	−4
11	−3	23	−4	−5	11	2
6	−4	30	−3	−3	6	0
5	−5	31	−3	−4	5	−2
8	−8	32	−5	−2	8	1
7	−1	33	−4	−7	7	−4

3.4.2 Основен алгоритъм

Алгоритъм 6 реализира бътерфлай алгоритъм върху сумата S и вектора $\chi(0)$, като в резултат S получава стойност $\left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right)\chi^{\mathrm{T}}$. При протичане на про-

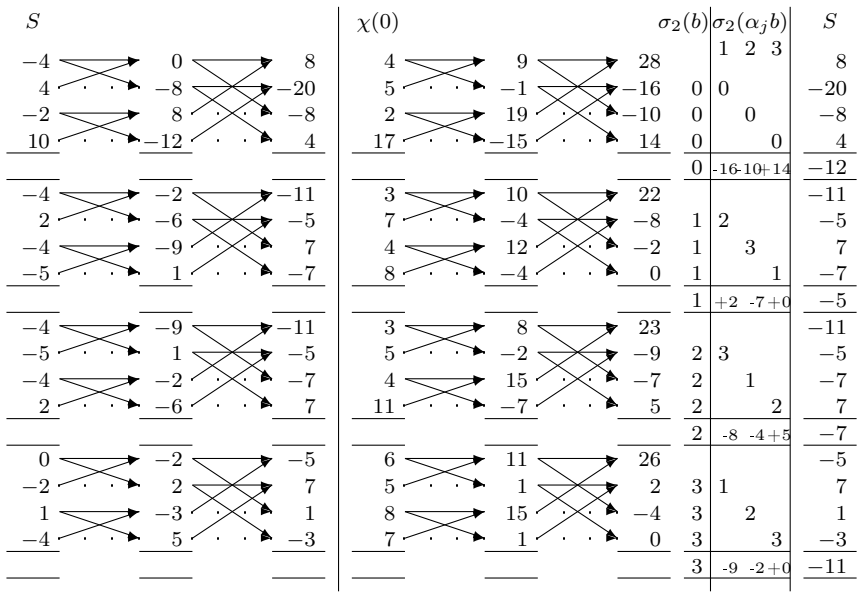
цедурата се търсят правилните места на неактивните координати. За целта се прилагат подходящи пермутации, реализирани с помощта на изображенията σ_l, $\nu^{(l)}$ и ν^{-1}. Бътерфлай алгоритмите за S и $\chi(0)$ са подобни на Алгоритъм 4. Не е нужно да се дефинират изображенията σ_1 и $\nu^{(1)}$, за да се появят в циклите при $k-l=1$ или $l=1$. В този случай се предполага, че съответните вектори $\sigma_1(b(1,i))$ и $\nu^{(1)}(u-1)$ са празни. Векторите $\mathbf{0}_l$ и $\mathbf{q}_l$ се състоят съответно от по l координати 0 и q. При това $\mathbf{0}_0$ е празен вектор.

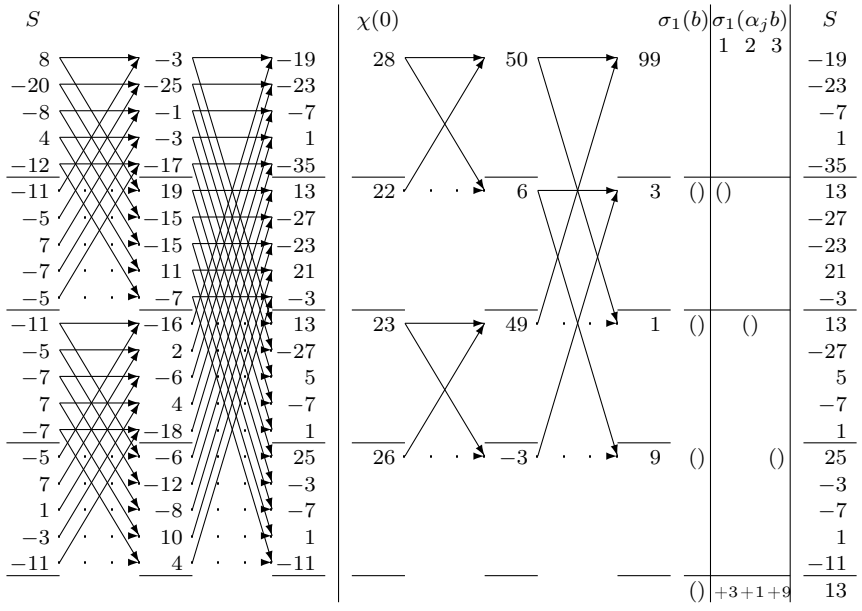
Алгоритъм 6 Основен алгоритъм

Вход: естествени числа q и k, масиви $\chi(0)$ и S с дължина $\theta(q,k)$
Изход: обновен масив S с дължина $\theta(q,k)$

for $l=1$ **to** $k-1$ **do**
  for $i=0$ **to** $q^{k-l-1}-1$ **do**
    for $l_1=1$ **to** m **do**
      for $i_1=0$ **to** $2^{m-l_1}-1$ **do**
        for $j_1=0$ **to** $2^{l_1-1}-1$ **do**
          for $u=1$ **to** $\theta(q,l)$ **do**
            $b'=(\sigma_{k-l}(b(k-l,i)), i_1 2^{l_1}+j_1, \nu^{(l)}(u-1))$;
            $u_1=\nu^{-1}(b')+1$; $u_2=u_1+2^{l_1-1}\theta(q,l)$;
            $A=S[u_1]$; $B=S[u_2]$;
            $S[u_1]=A+B$; $S[u_2]=A-B$;
          end for
          $b'=(\sigma_{k-l}(b(k-l,i)), i_1 2^{l_1}+j_1, \mathbf{0}_{l-1})$;
          $u_1=\nu^{-1}(b')+1$; $u_2=u_1+2^{l_1-1}\theta(q,l)$;
          $A=\chi(0)[u_1]$; $B=\chi(0)[u_2]$;
          $\chi(0)[u_1]=A+B$; $\chi(0)[u_2]=A-B$;
        end for
      end for
    end for
    for $j=1$ **to** $q-1$ **do**
      $b'=(\sigma_{k-l}(b(k-l,i)), j, \mathbf{0}_{l-1})$; $b''=(\sigma_{k-l}(\alpha_j b(k-l,i)), \mathbf{q}_l)$;
      $u_1=\nu^{-1}(b')+1$; $u_2=\nu^{-1}(b'')+1$; $S[u_2]=S[u_2]+\chi(0)[u_1]$;
    end for
  end for
end for

Пример 3.5. За кода от пример 3.2 Алгоритъм 6 дава следния резултат: $S=(-19,-23,-7,1,-35,13,-27,-23,21,-3,13,-27,5,-7,1,25,-3,-7,1,-11,13)$, като конкретните пресмятания са илюстрирани в схемите по-долу.





За получаване на тегловното разпределение се пресмята масивът wt по формулата

$$\mathrm{wt}[i] = \frac{3n - S[i]}{4} = \frac{297 - S[i]}{4}$$

за $i = 1, 2, \ldots, 21$. Така

$$\mathrm{wt} = (79, 80, 76, 74, 83, 71, 81, 80, 69, 75, 71, 81, 73, 76, 74, 68, 75, 76, 74, 77, 71).$$

Това означава, че тегловната функция на кода е

$$W(z) = 1 + 3(z^{68} + z^{69} + 3z^{71} + z^{73} + 3z^{74} + 2z^{75} + 3z^{76} + z^{77} + z^{79} + 2z^{80} + 2z^{81} + z^{83}).$$

3.4.3 Анализ на сложността

За пресмятане на $\chi(0)$ и $\chi(1)$ в Алгоритъм 5 се ползват $2\theta(q, k)$ събирания над $\mathbb{Z}$, откъдето сложността е $O(q^{k-1})$.

За определяне на новите позиции след пермутациите в Алгоритъм 5 се изискват $(q-1)q^{k-1}.(k-1)$ умножения в $\mathbb{F}_q$ и $(q-1)q^{k-1}$ събирания в $\mathbb{Z}$. Следва, че сложността на този етап е $O(kq^k)$.

Основният цикъл в Алгоритъм 6 има $k-1$ итерации за l. За първата бътерфлай процедура са нужни $q^{k-l-1}.m.2^{m-1}.\theta(q, l).4 = O(mq^{k-1})$ събирания в $\mathbb{Z}$. За пресмятанията над $\chi(0)$ се ползват $q^{k-l-1}.m.2^{m-1}.4 = O(mq^{k-l})$ събирания в $\mathbb{Z}$. За попълване на неактивните позиции се ползват $q^{k-l-1}.q.(k-l-1)$ умножения в $\mathbb{F}_q$ и $q^{k-l-1}.q$ събирания в $\mathbb{Z}$. Така общата сложност на Алгоритъм 6 е $O(kmq^{k-1})$.

Не е трудно да се види, че сложността на Алгоритъм 4 е $O(kmq^k)$. Освен това Алгоритъм 4 ползва масив с дължина q^k, докато Алгоритъм 6 ползва три масива, но с дължини $\theta(q, k) = (q^k - 1)/(q-1)$. Алгоритъм 6 е по-ефективен от досегашни алгоритми за пресмятане на тегловно разпределение, по-специално, когато дължината n или броят на елементите на полето q са големи.

Коментари към Глава 3

Описаният в раздел 1 подход е предложен от проф. Стефка Буюклиева и проф. Илия Буюклиев. Той е публикуван в [P3].

Резултатите, описани в останалите раздели тази глава, са получени в съавторство с проф. Илия Буюклиев и проф. Стефка Буюклиева. Те са поетапно докладвани [D5, D6, D12] и публикувани в [P2].

Глава 4

Пресмятане на радиус на покритие на линеен код над крайно поле чрез дискретни трансформации

В тази глава се обсъжда задачата за пресмятане на радиус на покритие на линеен $[n,k]_q$ код над крайно поле. Представен е алгоритъм, който се базира на преобразуването на Виленкин-Крестенсон. Преобразуването се прилага върху характеристичната функция на проверочната матрица на кода. Следствие 4.5 дава метод за пресмятане на радиуса на покритие чрез съкратеното разпределение на вектор с дължина $\theta(q, n-k)$. Методът е различен от предложения от Карповски за двоичния случай, представен в [52], където използваното преобразуване трябва да се направи $R(C)$ пъти. В разгледания алгоритъм това не е необходимо, ако е известна долна граница за радиуса на покритие R. Такава долна граница може да се получи по бърз евристичен алгоритъм [12]. Друго предимство на представения метод се състои в дължината на трансформирания вектор, която е $\theta(q, n-k)$, а не q^{n-k}, както е в теорема 4.3. Това прави методът удобен за прилагане за широк кръг от кодове.

За изясняване на връзката между материята в тази глава и разглежданите в предишните две глави алгоритми е нужно следващото понятие.

Дефиниция 4.1. Нека $b \in \mathbb{Z}^{\theta}$ е вектор с дължина $\theta(q,k)$ с целочислени координати. За всеки вектор-ред c на матрицата M_k нека е дефиниран векторът

$$c^{[b]_r} = (\mu_0 - \mu_1, \ldots, \mu_0 - \mu_{q-1}),$$

където $\mu_0, \mu_1, \ldots, \mu_{q-1}$ са координатите на $c^{[b]}$. Матрицата $M_k^{[b]_r}$ е съставена от векторите $c^{[b]_r}$, взети като редове. Сумата от стълбовете на $M_k^{[b]_r}$ се нарича *съкратено разпределение* на b и се означава с $r(b)$.

Матрицата $M_k^{[b]_r}$ е с размерност $\theta(q,k) \times (q-1)$, т. е. има $q^k - 1$ елемента. Очевидно

$$\left(c^{[b]_r}\right)^T = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & -1 & 0 \\ 1 & 0 & 0 & \dots & 0 & -1 \end{pmatrix} \cdot \left(c^{[b]}\right)^T$$

Лема 4.1. *Съкратеното разпределение $r(b)$ на вектора $b \in \mathbb{Z}^\theta$ е*

$$r(b) = [(q-1)J - q\mathcal{N}(M_k)]b^T,$$

където J е матрица с размерност $\theta \times \theta$, състояща се само от единици.

Доказателство. Нека $n = \sum_{u=1}^{\theta} b_u$. Ако c е i-тият ред на матрицата M_k, то i-тата координата на $\mathcal{N}(M_k)b^T$ е равна на

$$\sum_{\substack{1 \le u \le \theta \\ c_u \neq 0}} b_u = n - \mu_0,$$

където μ_0 е първият елемент на $c^{[b]}$. Нека c' е първият стълб на $M_k^{[b]}$. Тъй като n е сума от координатите на b, изпълнено е $\mathcal{N}(M_k)b^T = Jb^T - c'$, откъдето $c' = Jb^T - \mathcal{N}(M_k)b^T$.

От друга страна, за сумата от координатите на $c^{[b]_r}$ е в сила

$$(q-1)\mu_0 - \sum_{j=1}^{q-1} \mu_j = q\mu_0 - \sum_{j=0}^{q-1} \mu_u = q\mu_0 - \sum_{u=1}^{\theta(q,k)} b_u = q\mu_0 - n.$$

Следователно съкратеното разпределение на b е

$$r(b) = qc' - Jb^T = qJb^T - q\mathcal{N}(M_k)b^T - Jb^T = [(q-1)J - q\mathcal{N}(M_k)]\, b^T.$$

□

4.1 Пресмятане на радиус на покритие на линеен код над крайно просто поле

В този раздел се разглеждат само прости полета, поради което q е просто число и $\mathbb{F}_q = \mathbb{Z}_q = \{0, 1, \dots, q-1\}$. Теорема 4.3 дава връзката между преобразуването на Виленкин-Крестенсон и радиусът на покритие на линеен код.

За да се докаже основният резултат (теорема 4.3), е нужна следващата лема.

Лема 4.2. *Следното равенство е в сила за произволно* $x \in \mathbb{F}_q^s$*:*

$$\sum_{\omega \in \mathbb{F}_q^s} v_\omega(x) = \begin{cases} q^s, & \text{ако } x = \mathbf{0}, \\ 0, & \text{ако } x \neq \mathbf{0}. \end{cases}$$

Доказателство. Тази лема следва от [57, Chapter 5, Lemma 9], но тук е представено доказателство във въведените по-горе означения.

Тъй като $v_\omega(\mathbf{0}) = 1$, то $\sum_{\omega \in \mathbb{F}_q^s} v_\omega(\mathbf{0}) = q^s$.

В случая $x \neq \mathbf{0}$ доказателството е с индукция по s. В базовата стъпка $s = 1$ е изпълнено

$$\sum_{\omega \in \mathbb{F}_q} v_\omega(x) = \sum_{\omega=0}^{q-1} \xi^{x\omega} = \sum_{u=0}^{q-1} \xi^u = \frac{1-\xi^q}{1-\xi} = 0, \ \ \forall x \in \mathbb{F}_q \setminus \{0\}.$$

Нека равенството е в сила за някое естествено число s. Нека $x = (u, x')$, където $u \in \mathbb{F}_q$, $x' \in \mathbb{F}_q^s$, $x \neq \mathbf{0}$.

Ако $x' = \mathbf{0}$, но $u \neq 0$, е изпълнено

$$\sum_{\omega \in \mathbb{F}_q^{s+1}} v_\omega(x) = \sum_{i=0}^{q-1} (\xi^{ui} \sum_{\omega \in \mathbb{F}_q^s} v_\omega(\mathbf{0})) = q^s \sum_{i=0}^{q-1} \xi^i = q^s \frac{1-\xi^q}{1-\xi} = 0.$$

Ако $x' \neq \mathbf{0}$, то

$$\sum_{\omega \in \mathbb{F}_q^{s+1}} v_\omega(x) = \sum_{i=0}^{q-1} (\xi^{ui} \sum_{\omega \in \mathbb{F}_q^s} v_\omega(x')) = 0.$$

Съгласно принципа на математическата индукция равенството е в сила за всяко s. □

Нека C е линеен $[n, k]_q$ код с проверочна матрица H. Характеристичната функция на матрицата H се дефинира чрез

$$h_H(x) = \begin{cases} 1, & \text{ако } x \text{ е пропорционален на стълб от } H, \\ 0, & \text{в противен случай,} \end{cases} \tag{4.1}$$

където коефициентите на пропорционалност трябва да са различни от 0. Тази характеристична функция се използва за пресмятане на радиуса на покритие на кода. Следващата теорема е в сила за прости числа $q \geq 3$. Подобен резултат е публикуван [52, Theorem 2] за случая $q = 2$.

Теорема 4.3. *Нека C е линеен $[n,k]_q$ код с проверочна матрица H, където q е нечетно просто число, а $\widehat{h} : \mathbb{F}_q^{n-k} \to \mathbb{C}$ е трансформацията на Виленкин-Крестенсон на характеристичната функция $h = h_H$. Тогава радиусът на покритие $R(C)$ е равен на най-малкото естествено число t, такова че $\widehat{\widehat{h}}^t(y) \neq 0$ за всеки вектор $y \in \mathbb{F}_q^{n-k}$, $y \neq \mathbf{0}$.*

Доказателство. За степените на $\widehat{h}(\omega)$, $\omega \in \mathbb{F}_q^{n-k}$, е изпълнено:

$$\begin{aligned}\left(\widehat{h}(\omega)\right)^t &= \left(\sum_{x\in\mathbb{F}_q^{n-k}} h(x)v_\omega(x)\right)^t \\ &= \sum_{x_1,\ldots,x_t\in\mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t)v_\omega(x_1)\ldots v_\omega(x_t) \\ &= \sum_{x_1,\ldots,x_t\in\mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t)v_\omega(x_1+\ldots+x_t).\end{aligned}$$

След прилагане на трансформацията на Виленкин-Крестенсон за функцията $\widehat{h}^t$ се получава за $y \in \mathbb{F}_q^{n-k}\backslash\{\mathbf{0}\}$

$$\begin{aligned}\widehat{\widehat{h}}^t(y) &= \sum_{\omega\in\mathbb{F}_q^{n-k}} \left(\widehat{h}(\omega)\right)^t v_y(\omega) = \sum_{\omega\in\mathbb{F}_q^{n-k}} \left(\widehat{h}(\omega)\right)^t v_\omega(y) \\ &= \sum_{\omega\in\mathbb{F}_q^{n-k}} v_\omega(y) \sum_{x_1,\ldots,x_t\in\mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t)v_\omega(x_1+\ldots+x_t) \\ &= \sum_{x_1,\ldots,x_t\in\mathbb{F}_q^{n-k}} \sum_{\omega\in\mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t)v_\omega(x_1+\ldots+x_t+y) \\ &= \sum_{x_1,\ldots,x_t\in\mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t) \sum_{\omega\in\mathbb{F}_q^{n-k}} v_\omega(x_1+\ldots+x_t+y).\end{aligned}$$

Съгласно лема 4.2 е изпълнено $\sum_{\omega\in\mathbb{F}_q^{n-k}} v_\omega(x_1+\ldots+x_t+y) \neq 0$, само ако $x_1+\ldots+x_t+y = 0$. Това показва, че $\widehat{\widehat{h}}^t(y) \neq 0$ тогава и само тогава, когато съществува поне една редица $x_1,\ldots,x_t$ от (възможно повтарящи се) вектори от $\mathbb{F}_q^{n-k}$, така че $h(x_s) \neq 0$ за всяко x_s при $s = 1,\ldots,t$ и $x_1+\cdots+x_t = -y$. С други думи, $\widehat{\widehat{h}}^t(y) \neq 0$ тогава и само тогава, когато съществува редица $x_1,\ldots,x_t$ от вектори, пропорционални (с ненулев коефициент) на стълбове от матрицата H, такива че $x_1+\cdots+x_t = -y$. Нека n_y е броят на редиците $x_1,\ldots,x_t$ от вектори, пропорционални (с ненулев коефициент) на стълбове от H, чиято сума

е равна на $-y$. Тогава $\widehat{\widehat{h}}^t(y) = n_y q^{n-k}$ и $\widehat{\widehat{h}}^t(y) \neq 0$ тогава и само тогава, когато y може да бъде представено като линейна комбинация на най-много t стълба от проверочната матрица H.

Не е трудно да се види, че ако y може да се представи като сума на t (не непременно различни) стълба от $\widehat{H}$, то същият вектор може да се представи като сума на $t+1$ стълба. Последното твърдение не е изпълнено за $q = 2$. Наистина, ако $y = x_1 + \cdots + x_t$, то $y = x_1 + \cdots + x_{t-1} - \frac{q-1}{2}x_t - \frac{q-1}{2}x_t$. Следователно, ако $\widehat{\widehat{h}}^t(y) \neq 0$, то $\widehat{\widehat{h}^{t+1}}(y) \neq 0$.

Ако $t < R(C)$, то съществува вектор $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$, който не може да се представи като линейна комбинация на t стълба от H и тогава $\widehat{\widehat{h}}^t(y) = 0$. От друга страна, ако $t \geq R(C)$, то всеки вектор $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$ е линейна комбинация на не повече от t стълба на H и следователно $\widehat{\widehat{h}}^t(y) \neq 0$ за всяко $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$. □

Забележка 4.1. Ако $x_1, \ldots, x_t$ са пропорционални на стълбове в H и $x_1 + \cdots + x_t = y$, то съществува вектор $b \in \mathbb{F}_q^n$ с тегло $\mathrm{wt}(b) \leq t$, така че $y = Hb^T$. При това y е синдром на съседния клас $b + C$ и теглото на този съседен клас е не по-голямо от t.

Забележка 4.2. Същият метод може да се ползва за пресмятане на тегловното разпределение на лидерите на съседните класове на линеен код над $\mathbb{F}_q$ при нечетно просто q. Ако $t \geq 2$ е естествено число, то броят на лидерите на съседни класове с тегло t е равен на броя на векторите $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$, за които $\widehat{\widehat{h}}^t(y) \neq 0$ и $\widehat{\widehat{h}^{t-1}}(y) = 0$. Броят на лидерите на съседни класове с тегло 1 е равен на броя на ненулевите вектори $y \in \mathbb{F}_q^{n-k}$, за които $h(y) \neq 0$.

Някои свойства на пропорционалните вектори могат да бъдат използвани, за да се съкратят събиранията при изчисляване на трансформацията на Виленкин-Крестенсон, когато функцията $h : \mathbb{F}_q^s \to \mathbb{Z}$ приема целочислени стойности и удовлетворява равенството $h(x) = h(\alpha x)$ за всеки избор на $\alpha \in \mathbb{F}_q \backslash \{0\}$ и $x \in \mathbb{F}_q^s$. Пропорционалността е релация на еквивалентност в $\mathbb{F}_q^s$, която разделя линейното пространство на $\theta + 1$ класа на еквивалентност, където $\theta = \theta(q, s)$. Само класът $\{\mathbf{0}\}$ съдържа един елемент, а всеки от останалите класове се състои от $q - 1$ елемента.

Нека $g_1, \ldots, g_\theta$ са стълбовете на пораждащата матрица G_s на симплекс кода, дефинирана с (2.1). Изпълнено е

$$\widehat{h}(\mathbf{0}) = \sum_{x \in \mathbb{F}_q^s} h(x) v_{\mathbf{0}}(x) = \sum_{x \in \mathbb{F}_q^s} h(x) = h(\mathbf{0}) + (q-1) \sum_{u=1}^{\theta} h(g_u) \tag{4.2}$$

и

$$\begin{aligned}\widehat{h}(g_i) &= \sum_{x\in\mathbb{F}_q^s} h(x)v_{g_i}(x) = h(\mathbf{0}) + \sum_{u=1}^{\theta}\sum_{j=1}^{q-1} h(g_u)v_{g_i}(jg_u) \\ &= h(\mathbf{0}) + \sum_{u=1}^{\theta} h(g_u)\sum_{j=1}^{q-1}\left(\xi^{\langle g_i,g_u\rangle}\right)^j, \quad i=1,\ldots,\theta. \end{aligned} \tag{4.3}$$

Лема 4.4. *Нека q е нечетно просто число и $h:\mathbb{F}_q^s\to\mathbb{Z}$ е функция, за която $h(x)=h(\alpha x)$ за всеки избор на $\alpha\in\mathbb{F}_q\backslash\{0\}$ и $x\in\mathbb{F}_q^s$. Ако $\widehat{h}:\mathbb{F}_q^s\to\mathbb{C}$ е трансформацията на Виленкин-Крестенсон на h, то $\widehat{h}$ приема само целочислени стойности, при което $\widehat{h}(\omega)=\widehat{h}(\alpha\omega)$ за всеки избор на $\alpha\in\mathbb{F}_q\backslash\{0\}$ и $\omega\in\mathbb{F}_q^s$.*

Доказателство. Ако $\alpha\in\mathbb{F}_q$, $a\neq 0$, то

$$\widehat{h}(\alpha\omega)=\sum_{x\in\mathbb{F}_q^s} h(x)v_{\alpha\omega}(x)=\sum_{x\in\mathbb{F}_q^s} h(x)v_{\omega}(\alpha x)=\sum_{x\in\mathbb{F}_q^s} h(\alpha x)v_{\omega}(\alpha x)=\widehat{h}(\omega).$$

Последното равенство е изпълнено, понеже, ако x преминава множеството $\mathbb{F}_q^s$, то αx също преминава това множество за всяка фиксирана стойност на $\alpha\neq 0$.

За да се докаже, че $\widehat{h}$ приема само целочислени стойности, може да се ползват равенства (4.2) и (4.3). Очевидно, $\widehat{h}(\mathbf{0})\in\mathbb{Z}$. За $\widehat{h}(g_i)$, $i=1,\ldots,\theta$, е в сила

$$\sum_{j=1}^{q-1}\left(\xi^{\langle g_i,g_u\rangle}\right)^j = \begin{cases} q-1, & \text{ако } \langle g_i,g_u\rangle = 0, \\ -1, & \text{ако } \langle g_i,g_u\rangle \neq 0. \end{cases} \tag{4.4}$$

Окончателно, $h(\mathbf{0})$, $h(g_u)$ и $\sum_{j=1}^{q-1}\left(\xi^{\langle g_i,g_u\rangle}\right)^j$ в (4.3) са цели числа, откъдето стойностите на $\widehat{h}$ са цели числа. □

Ако h е характеристична функция на проверочна матрица H на линейния код C, дефинирана чрез (4.1), се получава следващото следствие.

Следствие 4.5. *Нека C е линеен $[n,k]_q$ код с проверочна матрица H, където q е нечетно просто число, а $\widehat{h}:\mathbb{F}_q^{n-k}\to\mathbb{C}$ е трансформацията на Виленкин-Крестенсон на характеристичната функция $h=h_H$. Тогава радиусът на покритие $R(C)$ е равен на най-малкото естествено число t, за което $\widehat{\widehat{h}^t}(g_i)\neq 0$ е в сила за всяко $i=1,\ldots,\theta(q,n-k)$.*

Доказателство. Очевидно $h(\alpha x) = h(x)$ за $\alpha \in \mathbb{F}_q \backslash \{0\}$, $x \in \mathbb{F}_q^{n-k}$. Освен това, характеристичната функция h на матрицата H има само целочислени стойности (0 и 1). Оттук с двукратно прилагане на Лема 4.4 се получава

$$\widehat{h}(\alpha\omega) = \widehat{h}(\omega) \quad \text{за } \alpha \in \mathbb{F}_q \backslash \{0\}, \omega \in \mathbb{F}_q^{n-k};$$

$$\widehat{\widehat{h}^t}(\alpha y) = \widehat{\widehat{h}^t}(y) \quad \text{за } \alpha \in \mathbb{F}_q \backslash \{0\}, y \in \mathbb{F}_q^{n-k}, t \in \mathbb{Z}, t \geq 1.$$

Последното показва, че всички различни стойности на $\widehat{\widehat{h}^t}(y)$ при $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$ и $t \in \mathbb{Z}$, $t \geq 1$, са измежду числата $\widehat{\widehat{h}^t}(g_i)$, $i = 1, \ldots, \theta(q, n-k)$. Сега резултатът следва от Теорема 4.3. □

Горното показва, че е достатъчно да се изчислят $\widehat{\widehat{h}^t}(\mathbf{0})$ и $\widehat{\widehat{h}^t}(g_i)$ за $i = 1, \ldots, \theta(q, n-k)$. От (4.3) и (4.4) следва, че

$$\widehat{h}(\alpha g_i) = \widehat{h}(g_i) = h(\mathbf{0}) + \sum_{u=1}^{\theta(q,s)} r_{iu} h(g_u),$$

където $\alpha \in \mathbb{F}_q \backslash \{0\}$ и

$$r_{iu} = \begin{cases} q-1, & \text{ако } \langle g_i, g_u \rangle = 0, \\ -1, & \text{ако } \langle g_i, g_u \rangle \neq 0. \end{cases}$$

С други думи, ако $\Lambda = (r_{iu})$ е матрица с размери $\theta(q,s) \times \theta(q,s)$, то

$$\begin{pmatrix} \widehat{h}(\mathbf{0}) \\ \widehat{h}(g_1) \\ \vdots \\ \widehat{h}(g_\theta) \end{pmatrix} = \left(\begin{array}{c|ccc} 1 & (q-1) & \ldots & (q-1) \\ \hline 1 & & & \\ \vdots & & \Lambda & \\ 1 & & & \end{array}\right) \cdot \begin{pmatrix} h(\mathbf{0}) \\ h(g_1) \\ \vdots \\ h(g_\theta) \end{pmatrix}$$

Матрицата Λ може да се получи от матрицата M_s със заместване на всички ненулеви елементи с -1 и всички нулеви елементи с $(q-1)$. Лесно се забелязва, че $\Lambda = (q-1)J - q\mathcal{N}(M_s)$, където J е $\theta(q,s) \times \theta(q,s)$ матрица, състояща се само от единици. Това означава, че Λ е трансформационната матрица на съкратеното разпределение на вектора $(h(g_1), \ldots, h(g_\theta))$ (съгласно лема 4.1). Ако $b = (h(g_1), \ldots, h(g_\theta))$, то

$$\begin{pmatrix} \widehat{h}(\mathbf{0}) \\ \widehat{h}(g_1) \\ \vdots \\ \widehat{h}(g_\theta) \end{pmatrix} = \begin{pmatrix} \widehat{h}(\mathbf{0}) \\ h(\mathbf{0}).\mathbf{1}^{\mathrm{T}} + \Lambda \cdot b^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} h(\mathbf{0}) + (q-1)\sum_{u=1}^{\theta} h(g_u) \\ \\ h(\mathbf{0}).\mathbf{1}^{\mathrm{T}} + r(b)^{\mathrm{T}} \end{pmatrix}. \qquad (4.5)$$

За пресмятане на $r(b)$ е приложим алгоритъмът, описан в Глава 2.

До края на раздела са дадени два примера. Първият показва усъвършенстването на изчисленията за трансформацията на Виленкин-Крестенсон. Вторият пресмята радиус на покритие на троичен линеен код по предложения метод.

Пример 4.1. За $q = 3$ и $s = 2$ функцията $h : \mathbb{F}_3^2 \to \mathbb{Z}$ се дефинира, както следва:

	0	0	0	1	1	1	2	2	2
x^T	0	1	2	0	1	2	0	1	2
$h(x)$	a	b	b	c	d	e	c	e	d

Тогава $TT(\widehat{h}) = V_2 \cdot TT(h)$, а именно

$$TT(\widehat{h}) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \xi & \xi^2 & 1 & \xi & \xi^2 & 1 & \xi & \xi^2 \\ 1 & \xi^2 & \xi & 1 & \xi^2 & \xi & 1 & \xi^2 & \xi \\ 1 & 1 & 1 & \xi & \xi & \xi & \xi^2 & \xi^2 & \xi^2 \\ 1 & \xi & \xi^2 & \xi & \xi^2 & 1 & \xi^2 & 1 & \xi \\ 1 & \xi^2 & \xi & \xi & 1 & \xi^2 & \xi^2 & \xi & 1 \\ 1 & 1 & 1 & \xi^2 & \xi^2 & \xi^2 & \xi & \xi & \xi \\ 1 & \xi & \xi^2 & \xi^2 & 1 & \xi & \xi & \xi^2 & 1 \\ 1 & \xi^2 & \xi & \xi^2 & \xi & 1 & \xi & 1 & \xi^2 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ b \\ c \\ d \\ e \\ c \\ e \\ d \end{pmatrix}$$

$$= \begin{pmatrix} a & +2b & +2c & +2d & +2e \\ a & -b & +2c & -d & -e \\ a & -b & +2c & -d & -e \\ a & +2b & -c & -d & -e \\ a & -b & -c & -d & +2e \\ a & -b & -c & +2d & -e \\ a & +2b & -c & -d & -e \\ a & -b & -c & +2d & -e \\ a & -b & -c & -d & +2e \end{pmatrix}.$$

Оттук за класовете пропорционални вектори се получават следните стойности:

x/ω	$h(x)$	$\widehat{h}(\omega)$				
00	a	a	$+2b$	$+2d$	$+2e$	$+2c$
01	b	a	$-b$	$-d$	$-e$	$+2c$
11	d	a	$-b$	$-d$	$+2e$	$-c$
21	e	a	$-b$	$+2d$	$-e$	$-c$
10	c	a	$+2b$	$-d$	$-e$	$-c$

Така трансформационната матрица е

$$\begin{pmatrix} 1 & 2 & 2 & 2 & 2 \\ 1 & -1 & -1 & -1 & 2 \\ 1 & -1 & -1 & 2 & -1 \\ 1 & -1 & 2 & -1 & -1 \\ 1 & 2 & -1 & -1 & -1 \end{pmatrix}, \text{ докато } \mathcal{N}(M_2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Пример 4.2. Нека C е линеен $[6,3]_3$ код с проверочна матрица

$$H = \begin{pmatrix} 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Изчисленията са представени в таблица 4.1. Следователно $R(C) = 3$. Тегловното разпределение на лидерите на съседните класове е: 6 лидера с тегло 1, 12 лидера с тегло 2 и 8 лидера с тегло 3.

Таблица 4.1: Изчисления към пример 4.2.

$x/\omega/y$	$h(x)$	$\widehat{h}(\omega)$	$\widehat{h}^2(\omega)$	$\widehat{\widehat{h}^2}(y)$	$\widehat{h}^3(\omega)$	$\widehat{\widehat{h}^3}(y)$
0 0 0	0	6	36	162	216	162
0 0 1	1	3	9	27	27	405
0 1 1	0	0	0	54	0	162
0 2 1	0	0	0	54	0	162
0 1 0	1	3	9	27	27	405
1 0 1	0	0	0	54	0	162
1 1 1	0	-3	9	0	-27	162
1 2 1	0	-3	9	0	-27	162
1 1 0	0	0	0	54	0	162
2 0 1	0	0	0	54	0	162
2 1 1	0	-3	9	0	-27	162
2 2 1	0	-3	9	0	-27	162
2 1 0	0	0	0	54	0	162
1 0 0	1	3	9	27	27	405

4.2 Пресмятане на радиус на покритие на линеен код над съставно крайно поле

В този раздел се разглеждат съставни полета, т. е. $q = p^m$, където p е просто число, $m \geq 2$ е естествено число и $\mathbb{F}_p = \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$. Резултатите от предишния раздел могат да бъдат преформулирани за съставни полета с използване на трансформацията на следите.

Нека C е линеен $[n,k]_q$ код с проверочна матрица H.

Теорема 4.6. *Нека C е $[n,k]_q$ код с проверочна матрица H, където $q = p^n$ за някое нечетно просто число p, а $\widehat{h} : \mathbb{F}_q^{n-k} \to \mathbb{C}$ е трансформацията на следите на характеристичната функция $h = h_H$. Тогава радиусът на покритие $R(C)$ е равен на най-малкото естествено число t, за което $\widehat{\widehat{h}^t}(y) \neq 0$ за всеки вектор $y \in \mathbb{F}_q^{n-k}$, $y \neq \mathbf{0}$.*

Доказателство. Както в доказателството на теорема 4.3, за $\omega, y \in \mathbb{F}_q^{n-k}$ се получава

$$\left(\widehat{h}(\omega)\right)^t = \sum_{x_1,\ldots,x_t \in \mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t)\tau_\omega(x_1 + \ldots + x_t) \tag{4.6}$$

и

$$\widehat{\widehat{h}^t}(y) = \sum_{x_1,\ldots,x_t \in \mathbb{F}_q^{n-k}} h(x_1)\ldots h(x_t) \sum_{\omega \in \mathbb{F}_q^{n-k}} \tau_\omega(x_1 + \ldots + x_t + y). \tag{4.7}$$

Съгласно лема 1.10 е изпълнено $\sum_{\omega \in \mathbb{F}_q^{n-k}} \tau_\omega(x_1 + \ldots + x_j + y) \neq 0$ точно тогава, когато $x_1 + \ldots + x_t + y = 0$. Така $\widehat{\widehat{h}^t}(y) \neq 0$ тогава и само тогава, когато y може да бъде представен като сума на t вектора (възможно повтарящи се), пропорционални с ненулев коефициент на стълбове от H.

Не е трудно да се види, че ако y може да се представи като сума на t (не непременно различни) вектора, пропорционални на стълбове в H, то същият вектор може да се представи като сума на $t+1$ вектора с това свойство. Действително, ако $y = x_1 + \ldots + x_t$, то $y = x_1 + \ldots + x_{t-1} - \frac{p-1}{2}x_t - \frac{p-1}{2}x_t$. Тук $p > 2$ е нечетно число. Следователно, ако $\widehat{\widehat{h}^t}(y) \neq 0$, то $\widehat{\widehat{h}^{t+1}}(y) \neq 0$.

В заключение, $\widehat{\widehat{h}^t}(y) \neq 0$ е изпълнено тогава и само тогава, когато y е линейна комбинация от не повече от t стълба на H. Ако $t < R(C)$, то съществува вектор $y \in \mathbb{F}_q^{n-k}$, който не е линейна комбинация на t стълба от H и тогава $\widehat{\widehat{h}^t}(y) = 0$. От друга страна, ако $t \geq R(C)$, то всеки вектор $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$ е линейна комбинация от най-много t стълба от H и следователно $\widehat{\widehat{h}^t}(y) \neq 0$ за всяко $y \in \mathbb{F}_q^{n-k} \backslash \{\mathbf{0}\}$. $\square$

Ако q е четно и някой вектор y е сума от t вектора, пропорционални на стълбове от H, то е възможно y да не е сума на $t+1$ вектора, пропорционални на стълбове от H. Но стойностите на $\widehat{\widehat{h}}^t(y)$ са неотрицателни поради (4.7) и лема 1.10. Тогава може да се разгледа сумата $\widehat{\widehat{h}}^1(y)+\cdots+\widehat{\widehat{h}}^t(y)$ вместо $\widehat{\widehat{h}}^t(y)$. Идеята е ползвана от Карповски за случая $q=2$ [52, Theorem 2].

Теорема 4.7. *Нека C е линеен $[n,k]_q$ код с проверочна матрица H, където $q=2^m$, а $\widehat{h}:\mathbb{F}_q^{n-k}\to\mathbb{C}$ е трансформацията на следите на характеристичната функция $h=h_H$. Нека*

$$\varphi_t(\omega)=\sum_{l=1}^{t}\left(\widehat{h}(\omega)\right)^l,\ \omega\in\mathbb{F}_q^{n-k},\quad t=1,\ldots,n,$$

и $\widehat{\varphi}_t:\mathbb{F}_q^{n-k}\to\mathbb{C}$ е трансформацията на следите на φ_t. Тогава радиусът на покритие $R(C)$ е равен на най-малкото естествено число t, за което $\widehat{\varphi}_t(y)\neq 0$ за всяко $y\in\mathbb{F}_q^{n-k}$, $y\neq\mathbf{0}$.

Лема 4.8. *Нека $q=p^m$ за някое просто число p и $h:\mathbb{F}_q^s\to\mathbb{Z}$ е функция, за която $h(x)=h(\alpha x)$ за всеки елемент $\alpha\in\mathbb{F}_q\backslash\{0\}$ и $x\in\mathbb{F}_q^s$. Ако $\widehat{h}:\mathbb{F}_q^s\to\mathbb{C}$ е трансформацията на следите на h, то $\widehat{h}$ е функция, която приема само целочислени стойности и $\widehat{h}(\omega)=\widehat{h}(\alpha\omega)$ за всеки $\alpha\in\mathbb{F}_q\backslash\{0\}$ и $\omega\in\mathbb{F}_q^s$.*

Доказателство. Ако $\alpha\in\mathbb{F}_q\backslash\{0\}$, то

$$\widehat{h}(\alpha\omega)=\sum_{x\in\mathbb{F}_q^s}h(x)\tau_{\alpha\omega}(x)=\sum_{x\in\mathbb{F}_q^s}h(x)\tau_{\omega}(\alpha x)=\sum_{x\in\mathbb{F}_q^s}h(\alpha x)\tau_{\omega}(\alpha x)=\widehat{h}(\omega),$$

поради свойствата на скаларното произведение над $\mathbb{F}_q$.

Нека $g_1,\ldots,g_\theta$ е максимално множество от ненулеви и непропорционални помежду си вектори от F_q^s, където $\theta=\theta(q,s)$. Тогава

$$\widehat{h}(\mathbf{0}) = \sum_{x\in\mathbb{F}_q^s}h(x)\tau_{\mathbf{0}}(x)=\sum_{x\in\mathbb{F}_q^s}h(x)=h(\mathbf{0})+(q-1)\sum_{u=1}^{\theta}h(g_u), \tag{4.8}$$

$$\begin{aligned}\widehat{h}(g_i) &= \sum_{x\in\mathbb{F}_q^s}h(x)\tau_{g_i}(x)=h(\mathbf{0})+\sum_{u=1}^{\theta}\sum_{\alpha\in\mathbb{F}_q\backslash\{0\}}h(\alpha g_u)\tau_{g_i}(\alpha g_u)\\ &= h(\mathbf{0})+\sum_{u=1}^{\theta}\sum_{\alpha\in\mathbb{F}_q\backslash\{0\}}h(g_u)\tau_{g_i}(\alpha g_u)\\ &= h(\mathbf{0})+\sum_{u=1}^{\theta}h(g_u)\sum_{\alpha\in\mathbb{F}_q\backslash\{0\}}\zeta^{\mathrm{Tr}(\alpha\langle g_i,g_u\rangle)},\ i=1,\ldots,\theta.\end{aligned} \tag{4.9}$$

Следователно функцията $\widehat{h}$ ще приема целочислени стойности, ако

$$\sum_{\alpha\in\mathbb{F}_q\setminus\{0\}} \zeta^{\mathrm{Tr}(\alpha\langle g_i,g_u\rangle)}$$

са цели числа за всеки $i,u=1,\ldots,\theta$.

Действително, ако $\langle g_i,g_u\rangle=0$, то $\alpha\langle g_i,g_u\rangle=0$ за всяко $\alpha\in\mathbb{F}_q\setminus\{0\}$ и сумата ще бъде $q-1$. Ако $\langle g_i,g_u\rangle\neq 0$, то $\{\alpha\langle g_i,g_u\rangle|\alpha\in\mathbb{F}_q,\alpha\neq 0\}=\mathbb{F}_q\setminus\{0\}$. След пресмятане на следите на елементите на това множество се получават по p^{m-1} стойности a за всяко $a\in\mathbb{F}_p\setminus\{0\}$ и $p^{m-1}-1$ стойности 0. Така

$$\sum_{\alpha\in\mathbb{F}_q\setminus\{0\}} \zeta^{\mathrm{Tr}(\alpha\langle g_i,g_u\rangle)}=\sum_{\alpha\in\mathbb{F}_q\setminus\{0\}} \zeta^{\mathrm{Tr}(\alpha)}=-1+p^{m-1}\sum_{a\in\mathbb{F}_p}\zeta^a=-1.$$

Оттук

$$\sum_{\alpha\in\mathbb{F}_q\setminus\{0\}} \zeta^{\mathrm{Tr}(\alpha\langle g_i,g_u\rangle)}=\begin{cases} q-1, & \text{ако } \langle g_i,g_u\rangle=0\\ -1, & \text{ако } \langle g_i,g_u\rangle\neq 0\end{cases} \tag{4.10}$$

С това доказателството е завършено. □

Равенствата (4.8), (4.9) и (4.10) позволяват да се използва съкратеното разпределение за пресмятане на радиус на покритие, като са приложими алгоритми, описани в предишните глави.

Коментари към Глава 4

Резултатите, описани в тази глава, са получени в съавторство с проф. Илия Буюклиев и проф. Стефка Буюклиева. Те са докладвани [D10] и приети за публикуване [P4].

Заключение

В дисертационния труд са представени решения на задачите за намиране на тегловно разпределение и радиус на покритие на линеен код над крайно поле. За целта пораждащата (проверочната) матрица е представена чрез характеристичен вектор, определящ броя на стълбовете, пропорционални с ненулев коефициент на стълбовете на специално избрана пораждаща матрица на симплекс кода. Разработени са алгоритми в зависимост от вида на крайното поле (просто или съставно). За основа на работата се ползват трансформациите на Уолш-Адамар, Виленкин-Крестенсон и следите, за които в литературата се споменава, че могат да се използват за целта. Благодарение на прехода към характеристичен вектор, предложените алгоритми имат по-малка сложност. Алгоритмите са значително ефективни при линейни кодове с големи дължини и при крайни полета с голям брой на елементите.

Научни приноси

Основни приносни моменти на дисертацията са:

1. Проучени и систематизирани са знанията за дискретните трансформации на Уолш-Адамар, Виленкин-Крестенсон и следите, като е показано приложението им за намиране на тегловно разпределение на линеен код.

2. Дефиниран е специален вид на пораждаща матрица на симплекс кода, който е удобен за определяне на характеристичен вектор на пораждаща (проверочна) матрица на линеен код. Тези дефиниции спомагат за получаване на естествени рекурентни връзки между трансформационните матрици от различните редове.

3. За линейни кодове над прости полета с характеристика $p > 2$, е разработен алгоритъм за намиране на тегловно разпределение по зададен характеристичен вектор, който има сложност $O(kp^{k+1})$, т. е. p пъти по-малка от сложността на известните досега алгоритми.

4. За линейни кодове над съставни крайни полета, е разработен общ алгоритъм за намиране на тегловно разпределение по зададен разширен характеристичен вектор, който използва трансформация на следите и самодуален базис, чрез който разглежданата трансформация се свежда до трансформация на Уолш-Адамар (при характеристика 2) или трансформация на Виленкин-Крестенсон. Сложността на този алгоритъм е $O(kmq^k)$.

5. За линейни кодове над съставни крайни полета, е разработен подобрен алгоритъм за намиране на тегловно разпределение по зададен характеристичен вектор, чрез който сложността се подобрява q пъти. Детайлно е описан този алгоритъм при съставни полета с характеристика 2.

6. Разработени са методи за намиране на радиус на покритие на линеен код над крайно поле (просто или съставно) по зададен характеристичен вектор на проверочната матрица, които са обобщение на предложения от Марк Карповски метод за двоични линейни кодове.

7. Разработените алгоритми са представени чрез теоретични обосновки, описания и схеми.

Изнесени доклади

[D1] Bouyukliev, Iliya, Stefka Bouyuklieva, Tatsuya Maruta, Paskal Piperkov. On the Computation of the Weight Spectrum of Linear Codes over Finite Fields. *Национален семинар по теория на кодирането „Професор Стефан Додунеков“*, Трявна, 11.11.2016.

[D2] Пиперков, Паскал, Илия Буюклиев, Стефка Буюклиева, Тадзя Марута. Изчисляване на тегловния спектър на линеен код над крайно поле с помощта на характеристичен спектър. *Семинар „Математически основи на информатиката“*, Велико Търново, 24.01.2017.

[D3] Piperkov, Paskal, Iliya Bouyukliev. On Walsh Transform and Matrix Factorization. *Optimal Codes and Related Topics (minisymposium in MDS2017)*, Sofia, 14.07.2017.

[D4] Piperkov, Paskal. Some Improvements on an Algorithm for Calculating Weight Spectrum of Linear Codes. *Национален семинар по теория на кодирането „Професор Стефан Додунеков“*, Чифлика, 02.12.2017.

[D5] Пиперков, Паскал. Трансформации от тип Фурие за намиране на тегловен спектър на линеен код над съставно крайно поле. *Семинар „Математически основи на информатиката“*, Велико Търново, 12.04.2018.

[D6] Piperkov, Paskal, Iliya Bouyukliev, Stefka Bouyuklieva. About Some Transforms in Coding Theory Related to Vilenkin-Chrestenson Transform. *International Colloquium on Differential Geometry and its Related Fields*, Veliko Tarnovo, 04.09.2018.

[D7] Piperkov, Paskal. Gaussian Binomial Coefficients. Its Applications in an Algorithm for Weight Spectrum of Linear Code. *Национален семинар по теория*

на кодирането „Професор Стефан Додунеков“, Велико Търново, 10.11.2018.

[D8] Piperkov, Paskal, Iliya Bouyukliev. Binary Representation of Vectors over Finite Prime Fields. *Ежегодна научна конференция на факултет „Математика и информатика“*, Велико Търново, 15.11.2019.

[D9] Piperkov, Paskal, Iliya Bouyukliev. How to Represent Vectors over Finite Fields for Fast Addition. *Национален семинар по теория на кодирането „Професор Стефан Додунеков“*, Чифлика, 22.11.2019.

[D10] Piperkov, Paskal, Iliya Bouyukliev. Some Methods for Calculating the Covering Radius of Linear Codes on the Base of Walsh-Hadamard and Vilenkin-Chrestenson Transforms. *Национален семинар по теория на кодирането „Професор Стефан Додунеков“*, Чифлика, 09.10.2020.

[D11] Piperkov, Paskal. Implementation through the Mathematica Package of Butterfly Algorithms for Determining the Parameters of a Linear Code. *Mathematical Software and Combinatorial Algoritms*, Sofia, 08.12.2020.

[D12] Пиперков, Паскал. Приложения на дискретни трансформации за пресмятане параметри на линеен код над съставно крайно поле. *Алгебрични и геометрични методи за защита на данни*, София, 09.02.2021.

[D13] Пиперков, Паскал. Функции и трансформация на Уолш. Математически основи и някои приложения. *STEMEDU 2021: Иновативно STEM образование*, Велико Търново, 26.04.2021.

Публикации по дисертацията

[P1] Bouyukliev, I., and Piperkov. P. On Walsh transform and matrix factorization. In *Eight International Workshop on Optimal Codes and Related Topics. Jul 10-14, 2017. Sofia, Bulgaria* (2017), pp. 55-60. ISSN 1313-1167.

[P2] Piperkov, P., Bouyukliev, I., and Bouyuklieva, S. An algorithm for computing the weight distribution of a linear code over composite finite field with characteristic 2. In *Recent Topics in Differential Geometry and its Related Fields,* T. Adachi and H. Hashimoto, Eds. World Scientific Publishing Company, 2019, pp. 163-181. ISBN 978-981-120-668-9. DOI:10.1142/9789811206696_0011.

[P3] Bouyukliev, I., Bouyuklieva, S., Maruta, T., and Piperkov, P. Characteristic vector and weight distribution of a linear code. *Cryptography and Communications 13*, 2 (2021), 263-282. ISSN 1936-2447. DOI:10.1007/s12095-020-00458-8.

[P4] Piperkov, P., Bouyukliev, I., and Bouyuklieva, S. An algorithm for computing the covering radius of a linear code based on Vilenkin-Chrestenson transform. In *New Horizons in Differential Geometry and its Related Fields,* T. Adachi and H. Hashimoto, Eds. World Scientific Publishing Company, 2022, pp. 105–123. ISBN 978-981-124-809-2. DOI:10.1142/9789811248108_0007.

Цитирания на публикации

[P3] Bouyukliev, I., Bouyuklieva, S., Maruta, T., and Piperkov, P. Characteristic vector and weight distribution of a linear code. *Cryptography and Communications 13*, 2 (2021), 263-282.

е цитирана в:

1. Pashinska, M., and Bouyukliev, I. A parallel algorithm for computing the weight spectrum of binary linear codes. In *Proceedings of the Seventeenth International Workshop on Algebraic and Combinatorial Coding Theory. ACCT 2020. 11–17 October 2020. on-line, Bulgaria* (2020), pp. 1–5. ISBN 978-1-6654-0287-3. DOI:10.1109/ACCT51235.2020.9383351.
2. Trofimiuk, G. A search method for large polarization kernels. In *2021 IEEE International Symposium on Information Theory. Proceedings. 12–20 July 2021. Virtual. Melbourne, Victoria, Australia* (2021), pp. 2084–2089. ISBN 978-1-5386-8210-4. DOI:10.1109/ISIT45174.2021.9517729.
3. Zhang, Q., Niu, G., Jiang, W., Qiao, M., and Li, D. Research on key index evaluation of power transmission and transformation wiring based on three-dimensional intelligent evaluation. *Energy Reports 8*, 7 (2022), 522–534. ISSN 2352-4847. DOI:10.1016/j.egyr.2022.05.089.

Л И Т Е Р А Т У Р А

[1] *Applications of Walsh functions. Proceedings* (Washington, D. C., 1970), vol. AD707431.

[2] *Applications of Walsh functions. Proceedings* (Washington, D. C., 1971), vol. AD727000.

[3] *Applications of Walsh functions. Proceedings* (Washington, D. C., 1972), vol. AD744650, Catholic University of America.

[4] *Applications of Walsh functions. Proceedings* (Washington, D. C., 1973), vol. AD763000.

[5] *Applications of Walsh functions. Proceedings* (Washington, D. C., 1974), vol. 74CH08615EMC.

[6] Ahmed, N., Rao, K. R., and Abdussattar, A. L. BEFORE and Hadamard transform. *IEEE Trans. Audio and Electroacoustics AU-19*, 3 (1971), 225–234.

[7] Andrews, H. C. *Computer Techniques in Image Processing.* Academic, New York, 1970.

[8] Andrews, H. C., and Caspari, K. L. A generalized technique for spectral analysis. *IEEE Trans. on Computers C-19*, 1 (1970), 16–25.

[9] Assmus, E. F., and Mattson, H. F. Coding and combinatorics. *SIAM Review 16*, 1 (1974), 349–388.

[10] Baart, R., Boothby, T., Cramwinckel, J., Fields, J., Joyner, D., Miller, R., Minkes, E., Roijackers, E., Ruscio, L., and Tjhai, C. GAP package GUAVA. https://www.gap-system.org/Packages/guava.html, 2019. Accessed: 2022-02-05.

[11] Baicheva, T. On the covering radius of ternary negacyclic codes with length up to 26. *IEEE Trans. Inform. Theory 47*, 1 (2001), 413–416.

[12] Baicheva, T., and Bouyukliev, I. On the least covering radius of binary linear codes of dimension 6. *Adv. Math. Commun. 4*, 3 (2010), 399–404.

[13] Berlecamp, E. R. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[14] Berlekamp, E., McEliece, R., and van Tilborg, H. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory 24*, 3 (1978), 384–386.

[15] Bespalov, M. S. Discrete Chrestenson transform. *Probl. Inf. Transm. 46*, 4 (2010), 353–375.

[16] Betten, A., Braun, M., Fripertinger, H., Kerber, A., Kohnert, A., and Wassermann, A. *Error-Correcting Linear Codes: Classification by Isometry and Applications.* Springer-Verlag, Berlin, 2006.

[17] Blahut, R. E. *Fast Algorithms for Signal Processing.* Cambridge University Press, Cambridge, 2010.

[18] Bogart, K., Goldberg, D., and Gordon, J. An elementary proof of the MacWilliams theorem on equivalence of codes. *Inform. and Control 37*, 1 (1978), 19–22.

[19] Bogdanova, G., and Kapralov, S. On the construction of q-ary Gray codes and their applications. In *Proc. of VII Intern. Workshop on Algebraic and Combinatorial Coding Theory, 18-24 June 2000, Bansko, Bulgaria* (2000), pp. 78–83.

[20] Bosma, W., Cannon, J., and Playoust, C. The Magma algebra system I: The user language. *J. Symbolic Computation 24*, 3–4 (1997), 235–265.

[21] Bouyukliev, I. What is Q-Extension? *Serdica J. Comput. 1*, 2 (2007), 115–130.

[22] Bouyukliev, I. The program WDHV v1.0 (a module in QextNewEdition). https://zenodo.org/record/3968198#.YpiUrDlBxH5, 2020. Accessed: 2022-06-02.

[23] Bouyukliev, I., and Bakoev, V. A method for efficiently computing the number of codewords of fixed weights in linear codes. *Discret. Appl. Math. 156*, 15 (2008), 2986–3004.

[24] Bouyukliev, I., and Bikov, D. Applications of the binary representation of integers in algorithms for Boolean functions. In *Mathematics and Education in Mathematics. Proceedings of the Forty Fourth Spring Conference of the Union of Bulgarian Mathematicians* (2015), pp. 161–166.

[25] Bouyukliev, I., and Gancheva, I. A method for calculation the weight distribution of the coset leaders of a linear code. In *ACCT'04, Kranevo, Bulgaria* (2004), pp. 67–72.

[26] Carlet, C. Boolean functions for cryptography and error-correcting codes. In *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Y. Crama and P. Hammer, Eds., vol. 134 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2010, pp. 257–397.

[27] Carlet, C. Vectorial Boolean functions for cryptography. In *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Y. Crama and P. Hammer, Eds., vol. 134 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2010, pp. 398–470.

[28] Chrestenson, H. E. A class of generalized Walsh functions. *Pacific J. of Math. 5*, 1 (1955), 17–31.

[29] Cohen, G., Honkala, I., Litsyn, S., and Lobstein, A. *Covering Codes*, vol. 54 of *North-Holland Mathematical Library*. Elsevier Science B.V., North-Holland, 1997.

[30] Cohen, G. D., Karpovsky, M. G., Mattson, H. F., and Schatz, J. R. Covering radius—survey and recent results. *IEEE Trans. Inform. Theory 31*, 3 (1985), 328–343.

[31] Cohen, G. D., Litsyn, S. N., Lobstein, A. C., and Mattson, H. F. Covering radius 1985-1994. *AAECC 8*, 3 (1997), 173–239.

[32] Cohen, G. D., Lobstein, A. C., and Sloane, N. J. A. Further results on the covering radius of codes. *IEEE Trans. Inform. Theory 32*, 5 (1986), 680–694.

[33] Coleman, R. P. Orthogonal functions for the logical design of switching circuits. *IRE Trans. on Electronic Computers EC-10* (1961), 379–383.

[34] Cooley, J. W., and Tukey, J. W. An algorithm for the machine calculation of complex Fourier series. *Math. Comp. 19* (1965), 297–301.

[35] Dodunekov, S. M. Some quasi-perfect double error correcting codes. *Problems Control Information Theory 15*, 5 (1986), 367–375.

[36] Dodunekov, S. M., Manev, K. N., and Tonchev, V. D. On the covering radius of binary $[14, 6]$ codes containing the all-one vector. *IEEE Trans. Inform. Theory 34*, 3 (1988), 591–593.

[37] Dodunekov, S. M., and Manev, N. L. An improvement of the Griesmer bound for some small minimum distances. *Discrete Appl. Math. 12*, 2 (1985), 103–114.

[38] Dodunekova, R. The duals of MMD codes are proper for error detection. *IEEE Trans. Inform. Theory 49*, 8 (2003), 2034–2038.

[39] Elliott, D. F., and Rao, K. R. *Fast Transforms. Algorithms, Analyses, Applications*. Academic Press, London, 1982.

[40] Farkov, Y. A. Discrete wavelets and the Vilenkin-Chrestenson transform. *Math. Notes 89* (2011), 871–884.

[41] Fine, N. J. On the Walsh functions. *Trans. Amer. Math. Soc. 65* (1949), 372–414.

[42] Good, I. J. The interaction algorithm and practical Fourier analysis. *J. of the Royal Stat. Soc., Ser. B. 20* (1958), 361–372.

[43] Graham, R. L., and Sloane, N. J. A. On the covering radius of codes. *IEEE Trans. Inform. Theory 31*, 3 (1985), 385–401.

[44] Gulliver, T., Bhargava, V., and Stein, J. Q-ary Gray codes and weight distributions. *Appl. Math. Comput. 103*, 1 (1999), 97–109.

[45] Hadamard, J. Résolution d'une question relative aux déterminants. *Bull. Sci. Math. (2) 17*, Part 1 (1893), 240–246.

[46] Harmuth, H. F. A generalized concept of frequency and some applications. *IEEE Trans. Inform. Theory IT-14*, 3 (1968), 375–382.

[47] Helleseth, T. On the covering radius of cyclic linear codes and arithmetic codes. *Discrete Applied Mathematics 11* (1985), 157–173.

[48] Henderson, K. W. Some notes on the Walsh functions. *IEEE Trans. Electron. Comput. EC-13*, Feb. (1964), 50–52.

[49] Huffman, W. C., and Pless, V. *Fundamentals of Error-Correcting Codes.* Cambridge Univ. Press, 2003.

[50] Joux, A. *Algorithmic Cryptanalysis.* Chapman and Hall/CRC, Boca Raton, FL 33487-2742, 2009.

[51] Karpovsky, M. G. On the weight distribution of binary linear codes. *IEEE Trans. Inform. Theory 25*, 1 (1979), 105–109.

[52] Karpovsky, M. G. Weight distribution of translates, covering radius, and perfect codes correcting errors of given weights. *IEEE Trans. Inform. Theory 27*, 4 (1981), 462–472.

[53] Karpovsky, M. G., Stanković, R. S., and Astola, J. T. *Spectral Logic and its Applications for the Design of Digital Devices.* John Wiley & Sons Ltd, 2008.

[54] Lechner, R. J. A transform theory for functions of binary variables. In *Theory of Switching*, vol. BL-30 of *Progress Rept.* Computation Lab., Harvard University, 1962, pp. 1–38.

[55] Lechner, R. J. Comment on "Computation of the fast Walsh-Fourier transform". *IEEE Trans. Comp. C-19* (1970), 174.

[56] Lidl, R., and Niederreiter, H. *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1986.

[57] MacWilliams, F. J., and Sloane, N. J. A. *The Theory of Error-Correcting Codes.* Elsevier Science Publishers, 1977.

[58] McLoughlin, A. The complexity of computing the covering radius of a code. *IEEE Trans. Inform. Theory 30*, 6 (1984), 800–804.

[59] Morgenthaler, G. W. On Walsh-Fourier series. *Trans. Amer. Math. Soc. 84* (1957), 472–507.

[60] Mullen, G. L., and Panario, D. *Handbook of Finite Fields*. Chapman and Hall/CRC, Boca Raton, FL 33487-2742, 2013.

[61] Paley, R. E. A. C. A remarkable series of orthogonal functions. *Proc. London Math. Soc. 34* (1932), 241–279.

[62] Pless, V. S., and Huffman, W. C. *Handbook on Coding Theory*. Elsevier Science B.V., 1998.

[63] Pratt, W. K., Kane, J., and Andrews, H. C. Hadamard transform image coding. *Proc. IEEE 57* (1969), 58–68.

[64] Seroussi, G., and Lempel, A. Factorization of symmetric matrices and trace-orthogonal bases in finite fields. *SIAM Journal on Computing 9*, 4 (1980), 758–767.

[65] Shanks, J. L. Computation of the fast Walsh-Fourier transform. *IEEE Trans. Comp. C-18*, 5 (1969), 457–459.

[66] Shum, F. Y. Y., Elliott, A. R., and Brown, W. O. Speech processing with walsh-hadamard transforms. *IEEE Trans. on Audio ans Electroacoustics AU-21*, 3 (1973), 174–179.

[67] Stanković, R. S., Astola, J. T., and Moraga, C. *Representation of Multiple-Valued Logic Functions*, vol. 37 of *Synthesis Lectures on Digital Circuits and Systems*. Morgan & Claypool, 2012.

[68] Stichtenoth, H. Subfield subcodes and trace codes. In *Algebraic Function Fields and Codes*, vol. 254 of *Graduate Texts in Mathematics*. Springer-Verlag, 2009, pp. 311–326.

[69] Sylvester, J. J. Thoughts on inverse orthogonal matrices, simultaneous sign-successions, and tesselated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *Phil. Mag. (4) 36* (1867), 461–475.

[70] Vilenkin, N. On a class of complete orthonormal systems. *Bull. Acad. Sci. URSS. Sér. Math. 11* (1947), 363–400.

[71] Walsh, J. L. A closed set of normal orthogonal functions. *American Journal of Math. 45*, 1 (1923), 5–24.