БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ПАСКАЛ НИКОЛАЕВ ПИПЕРКОВ

ДИСКРЕТНИ ТРАНСФОРМАЦИИ И ПРИЛОЖЕНИЕТО ИМ В ТЕОРИЯ НА КОДИРАНЕТО И КОМБИНАТОРИКАТА

автореферат на дисертация

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

София
2022 г.

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.

Увод

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 [11, 14, 26]. Обзор на бързи трасформации и техни приложения е направен в [5, 12, 19, 27].

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 [30]. Трансформацията на Уолш-Адамар се прилага при изследване на комбинаторни конфигурации като булеви и вектор булеви функции [8, 9], двоични линейни кодове [17] и други.

Функциите на Виленкин-Крестенсон [10, 29] 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 [17, 18].

За линейни кодове над съставни крайни полета е удобно да се ползва трансформацията на следите [1, 18]. 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. Теория на кодирането е систематически изградена, например в [2, 4, 15, 22, 24].

Повод за проучването са резултатите и идеите на Марк Карповски [17, 18] 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 [1]. 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 [18]. 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.

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“!

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 [21, 23].

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 $\mathrm{Tr}(\alpha)$.

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 ([25])**.** *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.

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. 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 1.7. 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.

Дефиниция 1.8. 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*.

Дефиниция 1.9. *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 $syn(x) = Hx^{\mathrm{T}} \in \mathbb{F}_q^{n-k}$.

Дефиниция 1.10. 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 1.2 ([15], 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.*

Дефиниция 1.11. *Линеен код с пълна дължина* е линеен код без нулеви стълбове в пораждащата матрица.

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 1.12. 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 $\mathcal{S}_{q,k}$.

1.3 Дискретна трансформация на Уолш-Адамар

Дефиниция 1.13 ([17])**.** 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

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

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:

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}, \; k > 1. \tag{2}$$

Теорема 1.3. *Ако f е булева функция на k променливи, то $W_f = H_k \cdot TT_f$.*

Съвсем естествено, дефиниция 1.13 може да се обобщи за псевдобулеви функции, т. е. функции от вида $f : \mathbb{F}_q^k \to \mathbb{Z}$, при което

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_2^k} f(x)(-1)^{\langle x,\omega \rangle}, \quad \omega \in \mathbb{F}_2^k. \tag{3}$$

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

$$\mathrm{wt}(\omega G)=\frac{n-\widehat{f}(\omega)}{2},\ \ \omega\in\mathbb{F}_2^k. \tag{4}$$

Този факт е споменат от Карповски [17] в случая, когато в G няма нулеви или повтарящи се стълбове, т. е. когато дуалният код е с минимално тегло, по-голямо от 2.

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 $s_1s_2\times t_1t_2$ матрицата

$$A\otimes B=\begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1t_1}B\\ a_{21}B & a_{22}B & \dots & a_{2t_1}B\\ \dots & & & \\ a_{s_11}B & a_{s_12}B & \dots & a_{s_1t_1}B\end{pmatrix}.$$

Кронекеровото произведение не е комутативно.

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

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

Гуд [14] 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 1.4. *Нека M е квадратна матрица от ред t и k е естествено число. Тогава*

$$\otimes^k M=B_1\cdot B_2\cdots B_k, \tag{5}$$

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.

Лехнер [20] прилага (5) за трансформацията на Уолш-Адамар, но в обратен ред на множителите, като споменава, че множителите комутират. Това е твърдението на следващата теорема.

Теорема 1.5. *Множителите в (5) комутират. Така няма значение редът на умножение.*

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:

$$V_1 = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{q-1} \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(q-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & \xi^{q-1} & \xi^{2(q-1)} & \dots & \xi^{(q-1)^2} \end{pmatrix}, \; V_{k+1} = V_1 \otimes V_k, \; k \in \mathbb{Z}, \, k \geq 1, \tag{6}$$

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.

Дефиниция 1.14. 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

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{Z}_q^k} f(x) v_\omega(x), \quad \omega \in \mathbb{Z}_q^k. \tag{7}$$

Подробна информация за тази трансформация, както и за други дискретни трансформации, свързани с преобразуването на Фурие, има например в [3, 13, 19].

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

$$\mathrm{wt}(\omega G) = \frac{(q-1)n - \widehat{f}(\omega)}{q}, \;\; \omega \in \mathbb{F}_q^k. \tag{8}$$

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 [18]. Той предлага използването на това преобразуване за изчисляване на тегловното разпределение на съседни класове.

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 [1, p. 367].

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

$$\tau_\omega(x) = \zeta^{\mathrm{Tr}(\langle \omega, x \rangle)} \tag{9}$$

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 (9) се дефинира преобразуване от тип Фурие [1], което се нарича трансформация на следите.

Дефиниция 1.15. 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

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_q^k} f(x)\tau_\omega(x) = \sum_{x \in \mathbb{F}_q^k} f(x)\zeta^{\mathrm{Tr}(\langle \omega, x \rangle)}, \quad \omega \in \mathbb{F}_q^k. \tag{10}$$

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

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

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 $k \in \mathbb{N}$.

Дефиниция 1.16. 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 $x \in \mathbb{F}_q^k$.

Забележка 1.1. Карповски [18] разглежда разширената матрица

$$G' = (\alpha_1 G | \alpha_2 G | \ldots | \alpha_{q-1} G) \tag{11}$$

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 1.6. *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*

$$\mathrm{wt}(\omega G) = \frac{(q-1)n - \widehat{f}(\omega)}{q}, \quad \omega \in \mathbb{F}_q^k, \tag{12}$$

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

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:

$$G_1 = (1)\,, \quad G_{k+1} = \begin{pmatrix} \mathbf{0} & \boldsymbol{\alpha_1} & \boldsymbol{\alpha_2} & \ldots & \boldsymbol{\alpha_{q-1}} & 1 \\ G_k & G_k & G_k & & G_k & \mathbf{0}^{\mathrm{T}} \end{pmatrix}\,,\ k \in \mathbb{N}. \tag{13}$$

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

$$\chi(C, G) = (\chi_1, \chi_2, \ldots, \chi_{\theta(q,k)}) \in \mathbb{Z}^{\theta(q,k)}, \tag{14}$$

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.

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 $\mathbb{Z}$), $i = 1, \ldots, \theta(q,k)$.*

От (13) се получава рекурентна връзка за матриците M_k, а именно: $M_1 = (1)$ и за всяко $k \in \mathbb{Z}, k \geq 2$

$$M_k = \begin{pmatrix} M_{k-1} & M_{k-1} & \ldots & M_{k-1} & \mathbf{0}^{\mathrm{T}} \\ M_{k-1} & M_{k-1} + J & \ldots & M_{k-1} + \alpha_{q-1} J & \mathbf{1}^{\mathrm{T}} \\ M_{k-1} & M_{k-1} + \alpha_2 J & \ldots & M_{k-1} + \alpha_2 \alpha_{q-1} J & \boldsymbol{\alpha_2}^{\mathrm{T}} \\ \vdots & & & & \\ M_{k-1} & M_{k-1} + \alpha_{q-1} J & \ldots & M_{k-1} + \alpha_{q-1}^2 J & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{1} & \ldots & \boldsymbol{\alpha_{q-1}} & 1 \end{pmatrix}. \qquad (15)$$

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

2.2 Характеристично разпределение

Дефиниция 2.2. Нека $\chi = (\chi_1, \ldots, \chi_t) \in \mathbb{Z}^t$ и $b = (b_1, \ldots, b_t) \in \mathbb{F}_q^t$, $t \in \mathbb{N}$. *Характеристично разпределение* на вектора b по отношение на χ е векторът

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

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

Дефиниция 2.3. 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 $B_1^{[\chi]}, \ldots, B_s^{[\chi]}$.

Теорема 2.2. *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.*

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

$$\chi=\left(\chi^{(0)}|\chi^{(1)}|\ldots|\chi^{(q-1)}|\chi^{(q)}\right), \tag{16}$$

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

$$\begin{pmatrix}
M_{k-1}^{[\chi^{(0)}]} & + & M_{k-1}^{[\chi^{(1)}]} & +\cdots+ & M_{k-1}^{[\chi^{(q-1)}]} & + & \mathbf{0}^{\mathrm{T}[\chi^{(q)}]} \\
M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_{q-1}}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \mathbf{1}^{\mathrm{T}[\chi^{(q)}]} \\
M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}_{\alpha_2}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_2\alpha_{q-1}}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \boldsymbol{\alpha_2}^{\mathrm{T}[\chi^{(q)}]} \\
 & & & \vdots & & & \\
M_{k-1}^{[\chi^{(0)}]} & + & \mathrm{SR}_{\alpha_{q-1}}(M_{k-1}^{[\chi^{(1)}]}) & +\cdots+ & \mathrm{SR}_{\alpha_{q-1}^2}(M_{k-1}^{[\chi^{(q-1)}]}) & + & \boldsymbol{\alpha_{q-1}}^{\mathrm{T}[\chi^{(q)}]} \\
\mathbf{0}^{[\chi^{(0)}]} & + & \mathbf{1}^{[\chi^{(1)}]} & +\cdots+ & \boldsymbol{\alpha_{q-1}}^{[\chi^{(q-1)}]} & + & 1^{[\chi^{(q)}]}
\end{pmatrix}, \tag{17}$$

където с SR е означена операцията *циклично преместване надясно.*

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.

Дефиниция 2.4. 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 $1\le l<k$ и векторът χ се разделя на $q+1$ части, както в (16), то $M_k^{[\chi]}(l)$ се пресмята по формулата

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

Матрицата $M_k^{[\chi]}(1)$ е с размерност $\theta(q,k) \times q$ и редове $M_1^{[\chi_u]}$, за $u = 1, \ldots, \theta(q,k)$. 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 $l = 1, \ldots, k-1$ е един и същ, именно $M_1^{[\chi^{(q)}]}$. Освен това, редът преди последния в $M_k^{[\chi]}(l)$ е един и същ за $l = 1, \ldots, k-2$. Действително, за всяко $l < k$ има някои редове, които се запазват във всички матрици $M_k^{[\chi]}(l')$ за $1 \leq l' < l$. Тези редове се наричат *неактивни редове*. Има точно $\theta(q, k-l)$ неактивни реда в $M_k^{[\chi]}(l)$, $l = 2, \ldots, k-1$.

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 $M_k^{[\chi]}(1)$, $M_k^{[\chi]}(2)$,..., $M_k^{[\chi]}(k-1)$, $M_k^{[\chi]}(k)$.

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

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

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

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

   където $\theta_0 = \theta(q, k-1)$ и $\theta_1 = \theta(q,k)$.

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 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.

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

$$T_{2,2} = \begin{pmatrix} I_q & I_q & I_q & \ldots & I_q & P_1^{-1} \\ I_q & P_1 & P_{\alpha_2} & \ldots & P_{\alpha_{q-1}} & I_q \\ I_q & P_{\alpha_2} & P_{\alpha_2^2} & \ldots & P_{\alpha_2\alpha_{q-1}} & P_{\alpha_2}P_1^{-1} \\ \vdots & & & & & \\ I_q & P_{\alpha_{q-1}} & P_{\alpha_{q-1}\alpha_2} & \ldots & P_{\alpha_{q-1}^2} & P_{\alpha_{q-1}}P_1^{-1} \\ E_0 & E_1 & E_2 & \ldots & E_{q-1} & E_1 \end{pmatrix}; \qquad (18)$$

  2. ако $k > l$, то

$$T_{k,l} = \begin{pmatrix} I_q \otimes T_{k-1,l} & O \\ O & I_q \end{pmatrix}; \qquad (19)$$

  3. ако $k = l > 2$, то

$$T_{k,k} =$$

$$\begin{pmatrix} I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes I_q & \dots & I_{\theta(q,k-1)} \otimes I_q & \mathbf{1} \otimes P_1^{-1} \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_1 & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}} & \mathbf{1} \otimes I_q \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_{\alpha_2} & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_2\alpha_{q-1}} & \mathbf{1} \otimes P_2 \cdot P_1^{-1} \\ \vdots & & & & \\ I_{\theta(q,k-1)} \otimes I_q & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}} & \dots & I_{\theta(q,k-1)} \otimes P_{\alpha_{q-1}^2} & \mathbf{1} \otimes P_{\alpha_{q-1}} \cdot P_1^{-1} \\ E_0 \;\; O & E_1 \;\; O & \dots & E_{q-1} \;\; O & I_q \end{pmatrix}. \tag{20}$$

Теорема 2.3. *Нека χ е характеристичен вектор на линеен $[n,k]_q$ код. Тогава*

$$\left(\widetilde{M}_k^{[\chi]}(l)\right)^{\mathrm{T}} = T_{k,l} \cdot \left(\widetilde{M}_k^{[\chi]}(l-1)\right)^{\mathrm{T}}, \quad l = 2, \dots, k, \tag{21}$$

и

$$\left(\widetilde{M}_k^{[\chi]}\right)^{\mathrm{T}} = T_{k,k} \cdot T_{k,k-1} \cdots T_{k,2} \cdot \widetilde{\chi}^{\mathrm{T}}. \tag{22}$$

2.4 Съкратено характеристично разпределение

Дефиниция 2.5. 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 2.4. *Ако $\chi \in \mathbb{Z}^t$ и $b \in \mathbb{F}_q^t$, $t \in \mathbb{N}$, то*

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

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

Лема 2.5. *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 $-n$.*

2.5 Сложност на алгоритмите и експериментални резултати

Общата сложност на описания алгоритъм 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 [16]. Според Жу, сложността на този алгоритъм, когато p се променя, е $O(kp^{k+2})$.

Представеният метод е програмиран на C/C++ [6]. За да се сравни ефективността, е ползвана програма на C, в която е вложен алгоритъмът, описан в [7], чиято сложност е като на алгоритмите, базирани на кодове на Грей. 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. Представени са и резултати за същите параметри, но получени с Magma V2.25-2 онлайн чрез Magma Calculator.

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

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 [28]. 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.

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*

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

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 1.16. Теорема 1.6 дава връзката между тегловното разпределение на C и

трансформацията на следите на f_G, която по дефиниция 1.15 е функцията

$$\widehat{f}(\omega) = \sum_{x \in \mathbb{F}_q^k} f_G(x)\tau_\omega(x) = \sum_{x \in \mathbb{F}_q^k} f_G(x)(-1)^{\mathrm{Tr}(\langle \omega, x \rangle)}, \quad \omega \in \mathbb{F}_q^k. \tag{23}$$

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 (2).

Горната лема показва, че

$$T_1 = H_m = \otimes^m H_1 = \otimes^m \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{24}$$

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

$$T_k = \otimes^k T_1 = \otimes^{km} H_1 = \otimes^{km} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{25}$$

и може да се ползва бътерфлай алгоритъм за пресмятане на трансформацията (23). Подобен псевдокод е описан от Жу [16, Algorithm 9.3].

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, индуктивно дефинирана по схемата в (13). 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

$$G_1 = (1), \quad G_k = \begin{pmatrix} \mathbf{0} & \boldsymbol{\alpha_1} & \dots & \boldsymbol{\alpha_{q-1}} & 1 \\ G_{k-1} & G_{k-1} & \dots & G_{k-1} & \mathbf{0}^{\mathrm{T}} \end{pmatrix}, \ k \in \mathbb{N}, \ k \geq 2. \tag{26}$$

Нека стълбовете на матрицата G_k са означени с g_u, $u = 1, \ldots, \theta(q,k)$, т. е.

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

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

$$\overline{G}_k = (0|\alpha_1 G_k| \ldots |\alpha_{q-1} G_k). \tag{27}$$

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 $\bar{g}_t$, $t = 1, \ldots, q^k$, т. е.

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

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 1.16. *Характеристичният вектор* $\chi = (\chi_1, \ldots, \chi_{\theta(q,k)})$ се дефинира чрез равенствата $\chi_u = f_G(g_u)$ за $u = 1, \ldots, \theta(q,k)$. *Разширеният характеристичен вектор* $\overline{\chi} = (\overline{\chi}_1, \ldots, \overline{\chi}_{q^k})$ се дефинира чрез равенствата $\overline{\chi}_t = f_G(\bar{g}_t)$ за $t = 1, \ldots, q^k$. От (27) 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 $\overline{\chi} = (0|\chi| \ldots |\chi)$.

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 $\widehat{\chi}^{\mathrm{T}} = \overline{P}_k \cdot \overline{\chi}^{\mathrm{T}}$.

$$\overline{P}_k \cdot \overline{\chi}^{\mathrm{T}} = \begin{pmatrix} (q-1)\sum_{u=1}^{\theta(q,k)} \chi_u \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \\ \vdots \\ \left(\sum_{j=1}^{q-1} P_{k,\alpha_j}\right) \chi^{\mathrm{T}} \end{pmatrix}. \tag{28}$$

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 $\alpha \in \mathbb{F}_q \backslash \{0\}$.

$$P_k = \begin{pmatrix} & & \mathbf{\Lambda_0}^{\mathrm{T}} \\ & T_1 \otimes P_{k-1} & \mathbf{\Lambda_{\alpha_1}}^{\mathrm{T}} \\ & & \vdots \\ & & \mathbf{\Lambda_{\alpha_{q-1}}}^{\mathrm{T}} \\ \mathbf{\Lambda_0} & \mathbf{\Lambda_{\alpha_1}} \quad \ldots \quad \mathbf{\Lambda_{\alpha_{q-1}}} & \Lambda_1 \end{pmatrix}, \tag{29}$$

където $\Lambda_\alpha = (-1)^{\mathrm{Tr}(\alpha)}$ за $\alpha \in \mathbb{F}_q$, а $\mathbf{\Lambda_\alpha}$ е вектор ред с дължина по подразбиране и едни и същи координати Λ_α. Последното равенство и (24) 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

$$\chi = (\chi^{(0)} | \chi^{(1)} | \ldots | \chi^{(q-1)} | \chi_{\theta(q,k)}), \tag{30}$$

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

Нека $Num(\alpha) \in \{0, 1, \ldots, q-1\}$ е номерът на елемента α в наредбата на полето $\mathbb{F}_q$, т. е. $\alpha_{Num(\alpha)} = \alpha$. Изпълнено е

$$P_k \cdot \chi^{\mathrm{T}} = \begin{pmatrix} & \begin{pmatrix} P_{k-1} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{k-1} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{k-1} \cdot {\chi^{(Num(1))}}^{\mathrm{T}} + \chi_\theta . \mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1} \cdot {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ (T_1 \otimes I_{\theta(q,k-1)}) \cdot & \\ & \\ \Lambda_0 \sum \chi^{(0)} + \cdots + \Lambda_1 \left(\chi_\theta + \sum \chi^{(Num(1))}\right) + \cdots + \Lambda_{\alpha_{q-1}} \sum \chi^{(q-1)} \end{pmatrix}, \tag{31}$$

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

За $1 < l < k$ е в сила

$$P_l \cdot {\chi'}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} = \begin{pmatrix} \big(T_1 \otimes I_{\theta(q,l-1)}\big) \cdot \begin{pmatrix} P_{l-1} \cdot {\chi'^{(0)}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ P_{l-1} \cdot {\chi'^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{l-1} \cdot {\chi'^{(Num(1))}}^{\mathrm{T}} + \chi'_{\theta(q,l-1)}.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{l-1} \cdot {\chi'^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0\big(\chi_\theta + \sum \chi'^{(0)}\big) + \Lambda_{\alpha_1} \sum \chi'^{(1)} + \ldots \end{pmatrix}, \quad (32)$$

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 Λ_1. 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 $\chi'^{(j)}$, $j = 0, \ldots, q-1$.

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

$$P_{k,\alpha} \cdot \chi^{\mathrm{T}} = \begin{pmatrix} (\Lambda^{(\alpha)} \otimes I_{\theta(q,k-1)}) \cdot \begin{pmatrix} P_{k-1,\alpha} \cdot {\chi^{(0)}}^{\mathrm{T}} \\ P_{k-1,\alpha} \cdot {\chi^{(1)}}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha} \cdot {\chi^{(Num(1))}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha} \cdot {\chi^{(q-1)}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0 \sum \chi^{(0)} + \Lambda_{\alpha\alpha_1} \sum \chi^{(1)} + \cdots + \Lambda_\alpha \left(\chi_\theta + \sum \chi^{(Num(1))}\right) + \cdots \end{pmatrix}. \quad (33)$$

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

$$P_{k,\alpha}\cdot\chi^{\mathrm{T}} = \begin{pmatrix} (T_1\otimes I_{\theta(q,k-1)})\cdot \begin{pmatrix} P_{k-1,\alpha}\cdot {\chi^{(\pi_\alpha^{-1}(0))}}^{\mathrm{T}} \\ P_{k-1,\alpha}\cdot {\chi^{(\pi_\alpha^{-1}(1))}}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha}\cdot {\chi^{(\pi_\alpha^{-1}(Num(\alpha)))}}^{\mathrm{T}} + \chi_\theta.\mathbf{1}^{\mathrm{T}} \\ \vdots \\ P_{k-1,\alpha}\cdot {\chi^{(\pi_\alpha^{-1}(q-1))}}^{\mathrm{T}} \end{pmatrix} \\ \Lambda_0\sum\chi^{(0)} + \Lambda_{\alpha\alpha_1}\sum\chi^{(1)} + \cdots + \Lambda_\alpha\left(\chi_\theta + \sum\chi^{(Num(1))}\right) + \cdots \end{pmatrix}, \tag{34}$$

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

Сравнението на равенства (31) и (34) 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. 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.

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 $0 \le z \le \theta(q,k)-1$ и $\rho(z) = (\rho_1,\ldots,\rho_{k-1})$, то

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

Лема 3.5. *Ако* $0 \le z \le \theta(q,k)-1$ *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 $0 \le \rho_t \le q$, $t = 1,\ldots,k-1$.

Следствие 3.6. *Изображението ρ е инективно.*

Лема 3.7. *Ако $0 \leq z \leq \theta(q,k) - 1$, $\rho(z) = (\rho_1, \ldots, \rho_{k-1})$ 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 $\rho_{t+1} = \cdots = \rho_{k-1} = 0$.*

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.

В разработеният алгоритъм се ползват три масива с дължина $\theta(q,k)$, означени с $\chi(0)$, $\chi(1)$ и S. 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. Масивът S служи за формиране на вектора $SP(\chi)$.

3.4.2 Основен алгоритъм

Реализира се 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 процедурата 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 алгоритъма, описан в раздел 3.2.

3.4.3 Анализ на сложността

Общата сложност на подобрения алгоритъм е $O(kmq^{k-1})$.

Не е трудно да се види, че сложността на алгоритъма от раздел 3.2 е $O(kmq^k)$. Освен това се ползва масив с дължина q^k, докато подобреният алгоритъм ползва три масива, но с дължини $\theta(q,k) = (q^k - 1)/(q-1)$. Описаният подобрен алгоритъм 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.

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

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 *свкратено разпределение* на b и се означава с $r(b)$.

Лема 4.1. *Свкратеното 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 $\theta \times \theta$, свстояща се само от единици.

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 $\mathbb{F}_q = \mathbb{Z}_q = \{0, 1, \ldots, q-1\}$.

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

$$h_H(x) = \begin{cases} 1, & \text{ако } x \text{ е пропорционален на стълб от } H, \\ 0, & \text{в противен случай,} \end{cases} \tag{35}$$

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 [18, Theorem 2] за случая $q = 2$.

Теорема 4.2. *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 $\widehat{h^t}(y) \neq 0$ за всеки вектор $y \in \mathbb{F}_q^{n-k}$, $y \neq \mathbf{0}$.*

Забележка 4.1. 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 $\widehat{h^t}(y) \neq 0$ и $\widehat{h^{t-1}}(y) = 0$. 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 $h(y) \neq 0$.

Нека $g_1, \ldots, g_\theta$ са стълбовете на пораждащата матрица G_s на симплекс кода, дефинирана с (13). Изпълнено е

$$\widehat{h}(\mathbf{0}) = \sum_{x \in \mathbb{F}_q^s} h(x) v_{\mathbf{0}}(x) = \sum_{x \in \mathbb{F}_q^s} h(x) = h(\mathbf{0}) + (q-1) \sum_{u=1}^{\theta} h(g_u) \tag{36}$$

и

$$\begin{aligned} \widehat{h}(g_i) &= \sum_{x \in \mathbb{F}_q^s} h(x) v_{g_i}(x) = h(\mathbf{0}) + \sum_{u=1}^{\theta} \sum_{j=1}^{q-1} h(g_u) v_{g_i}(j g_u) \\ &= h(\mathbf{0}) + \sum_{u=1}^{\theta} h(g_u) \sum_{j=1}^{q-1} \left(\xi^{\langle g_i, g_u \rangle}\right)^j, \quad i = 1, \ldots, \theta. \end{aligned} \tag{37}$$

Лема 4.3. *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 $\omega \in \mathbb{F}_q^s$.*

Следствие 4.4. *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 $i = 1, \ldots, \theta(q, n-k)$.

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

$$\sum_{j=1}^{q-1} \left(\xi^{\langle g_i, g_u \rangle}\right)^j = \begin{cases} q-1, & \text{ако } \langle g_i, g_u \rangle = 0, \\ -1, & \text{ако } \langle g_i, g_u \rangle \neq 0, \end{cases} \tag{38}$$

от (37) 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

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

Ако $b = (h(g_1), \ldots, h(g_\theta))$, то

$$\begin{pmatrix} \widehat{h}(\mathbf{0}) \\ \widehat{h}(g_1) \\ \vdots \\ \widehat{h}(g_\theta) \end{pmatrix} = \begin{pmatrix} \widehat{h}(\mathbf{0}) \\ h(\mathbf{0}).\mathbf{1}^{\mathrm{T}} + \Lambda \cdot b^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} h(\mathbf{0}) + (q-1)\sum_{u=1}^{\theta} h(g_u) \\ \\ h(\mathbf{0}).\mathbf{1}^{\mathrm{T}} + r(b)^{\mathrm{T}} \end{pmatrix}. \tag{39}$$

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

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.

Теорема 4.5. *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 $y \in \mathbb{F}_q^{n-k}$, $y \neq \mathbf{0}$.*

Теорема 4.6. *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 4.7. *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 $\omega\in\mathbb{F}_q^s$.*

Отново може да се използва съкратеното разпределение за пресмятане на радиус на покритие, като са приложими алгоритми, описани в предишните глави.

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.

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.

Литература

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

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

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

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

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

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

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

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

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

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

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

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

[18] 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.

[19] 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.

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

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

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

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

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

[25] 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.

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

[27] 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.

[28] 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.

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

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