БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

МАРИЯ СТЕФАНОВА ДЖУМАЛИЕВА-СТОЕВА

АЛГОРИТМИ ЗА ИЗСЛЕДВАНЕ НА КОМБИНАТОРНИ СТРУКТУРИ

Автореферат на

Д И С Е Р Т А Ц И Я

за присъждане на научната степен
"доктор"
по научна специалност
01.01.12 – Информатика

Велико Търново
2015 г.

Дисертацията съдържа 124 страници. Библиографията обхваща 83 заглавия, от които 5 на кирилица и 78 на латиница.

Дисертационната работа е обсъдена и насочена за защита от обединено научно звено от хабилитирани специалисти на ИМИ-БАН, НБУ и ФМИ - Софийски университет, на 30 март 2015г. (дата на вътрешна защита).

Изследванията по дисертационната работа са извършвани в Института по Математика и Информатика при БАН.

Защитата на дисертацията ще се състои на ..... г. от ..... ч. в зала ..... на .....

Материалите по защитата са на разположение в библиотеката на Института по математика и информатика, ул. „Акад. Г. Бончев“, бл. 8, София.

Автор: Мария Стефанова Джумалиева-Стоева

Научен ръководител: проф. дмн. Илия Георгиев Буюклиев

Заглавие: Алгоритми за изследване на комбинаторни структури

С Ъ Д Ъ Р Ж А Н И Е

Увод

В тази работа разглеждаме алгоритми за решаване на основните задачи, свързани с комбинаторни структури - конструиране и класификация. Представени са алгоритми за генериране с отхвърляне на изоморфните обекти, които решават едновременно посочените две задачи.

Основните обекти, които разглеждаме и за които представяме методи за конструиране и класификация са линейните кодове и в частност двоичните самоортогонални и самодуални кодове. Кодовете се разглеждат в термините на пораждащи матрици, което ги прави комбинаторни структури.

Линейни кодове

Нека $\mathbb{F}_q$ е крайно поле с q елемента, а $\mathbb{F}_q^n$ е n-мерното векторно пространство над $\mathbb{F}_q$. Тегло по Хеминг $\mathrm{wt}(x)$ на вектора $x = (x_1, x_2, ..., x_n) \in \mathbb{F}_q^n$ наричаме броя на ненулевите му координати $\mathrm{wt}(x) = |\{i|x_i \neq 0\}|$, а разстояние по Хеминг $d(x, y)$ между два вектора $x = (x_1, x_2, ..., x_n)$ и $y = (y_1, y_2, ..., y_n)$ от $\mathbb{F}_q^n$ се нарича броят на координатите, в които те се различават $d(x, y) = |\{i|x_i \neq y_i\}|$.

Нека $(u, v) = u \cdot v = \sum_{i=1}^{n} u_i v_i$ е стандартното евклидово скаларно произведение над $\mathbb{F}_q$. Два вектора наричаме ортогонални, ако тяхното скаларно произведение е нула.

Всяко k-мерно подпространство C на $\mathbb{F}_q^n$ наричаме линеен код с дължина n и размерност k и казваме, че е q-ичен $[n, k]$ код, или $[n, k]_q$ код. Минимално разстояние $d(C)$ на кода C се нарича най-малкото от разстоянията между две различни кодови думи, а минимално тегло на кода C наричаме най-малкото от всички ненулеви тегла в кода. Ако d е минималното разстояние на кода казваме, че C е $[n, k, d]_q$ код. Ако $q = 2$ казваме, че C е двоичен линеен код. В тази работа разглеждаме само двоични кодове.

Пораждаща матрица на линеен $[n, k]_q$ код C наричаме всяка матрица с k реда и n стълба с елементи от полето $\mathbb{F}_q$, чиито редове образуват базис на C.

Два двоични линейни $[n, k]$ кода са еквивалентни, ако кодовите думи на единия могат да се изобразят в кодовите думи на другия чрез пермутация на координатни позиции. Автоморфизъм на двоичен линеен код C наричаме пермутация, която изобразява кодовите думи на кода отново в кодови думи от C. Всички автоморфизми на един код образуват група, която наричаме група от автоморфизми на кода C и означаваме с $\mathrm{Aut}(C)$. В двоичния случай $\mathrm{Aut}(C)$ е подгрупа на симетричната група S_n.

Ортогоналното допълнение $C^{\perp} = \{v \in \mathbb{F}_q^n | (u, v) = 0, \forall u \in C\}$ на кода C относно евклидовото скаларно произведение наричаме ортогонален (дуален)

код на кода C. Един линеен код се нарича самоортогонален, ако $C \subseteq C^{\perp}$ и самодуален, когато $C = C^{\perp}$.

Конструиране и класификация на комбинаторни обекти

Комбинаторни обекти конструираме (генерираме) при зададени правила от елементите на едно или няколко крайни множества от обекти (букви на крайна азбука или вектори над крайна азбука). Нека са генерирани краен брой комбинаторни структури, които са включени в множество A и дефинираме релация на еквивалентност над това множество - $R_{\cong} \subseteq A \times A$ или $(A, \cong)$. Релация на еквивалентност наричаме релация, която е рефлексивна, симетрична и транзитивна. Релацията $(A, \cong)$ разбива множеството A на класове на еквивалентност $[a_i] = \{b | b \cong a\}$. Да се определи дали два обекта са еквивалентни означава да се определи дали те принадлежат на един и същ клас на еквивалентност и е прието да се нарича проблем за изоморфизъм. Задачата за класификация се състои в това да бъде намерен точно по един елемент от всеки клас на еквивалентност. Много често се използва действие на група върху множество, което дефинира релация на еквивалентност в него. Действие на група G върху множеството A наричаме изображение $\varphi : G \times A \to A$, което удовлетворява следните условия: за всяко $X \in A$ имаме $1_G * X = X$ и за всеки $g_1, g_2 \in G$ е изпълнено $g_1.g_2 * X = g_2 * (g_1 * X)$. При такова действие, множеството A се разбива на орбити, като два елемента X и Y принадлежат на една и съща орбита ако съществува елемент на групата g, такъв че $g * X = Y$. В такъв случай класовете на еквивалентност са орбитите при това действие, а два елемента са еквивалентни, когато принадлежат на една и съща орбита.

Проблем за изоморфизъм на комбинаторни структури

Основно място във всеки алгоритъм за класификация, а също и в алгоритмите за генериране на комбинаторни обекти, заема проблемът за изоморфизъм. Термините еквивалентност и изоморфизъм са сходни и се употребяват в зависимост от конкретната дефиниция и свойствата на конкретната структура. Изоморфизмът се дефинира чрез биективно изображение между два обекта (например графи, двоични матрици), което от своя страна дефинира релация на еквивалентност между тези обекти.

Съществуват два основни типа алгоритми, които решават проблема за изоморфизъм между X и Y. Първият тип алгоритми са специфични за конкретната структура и обектите се сравняват чрез инварианти. Под инварианта разбираме функция $f : A \to J$, която изобразява един обект от множеството A в число или вектор. Обектите от един и същ клас на еквивалентност имат една и съща стойност на инварианта. Инвариантните функции се явяват необходимо, но не и достатъчно условие за еквивалентност между обекти,

т.е. ако $f(X) \neq f(Y)$, то X и Y не са изоморфни, а в противен случай, ако $f(X) = f(Y)$, са необходими и други изчисления за достигане на конкретен отговор на задачата за изоморфизъм на двата обекта.

Ако разглеждаме класовете на еквивалентност в множеството A като орбити, получени от действието на група G над множеството, то ефективността на този тип специфични алгоритми до голяма степен зависи от реда на групата G. При големи групи ефективността значително намалява.

Алгоритми, базирани на подхода за сравнение чрез инварианти, са разработени за линейни кодове [18, 28, 39], дизайни [32], адамарови матрици [29].

Вторият тип алгоритми се състои в намирането и сравняването на каноничните форми на обектите X и Y, като за целта се дефинира подходящо канонично изображение. Канонично изображение наричаме функцията $\varphi : A \to \Omega$, за която е изпълнено, че $\varphi(X) = \varphi(Y)$, т. и с. т., когато $X \cong Y$ и че $X \cong \varphi(X)$. Обектът X е в канонична форма, ако той съвпада с образа си спрямо каноничното изображение φ. Този подход е използван за линейни кодове [7], за дизайни и за графи [26, 34, 27]. В повечето случаи този тип алгоритми са по-ефективни от специфичните алгоритми от предходния тип.

В тази работа изучаваме и представяме подход, при който се използват фундаментални комбинаторни обекти, чрез които повечето структури могат да бъдат представени. С други думи проблемът за изоморфизъм на даден тип обекти се свежда до изоморфизъм на основни обекти като графи и двоични матрици. Примери за това как комбинаторни структури могат да бъдат представени като графи са дадени от Kaski и Östergård [25]. Освен това съществуват алгоритми за изоморфизъм на графи [16, 17, 23, 24, 30, 31]. Най-ефективният алгоритъм за изоморфизъм на графи, известен до този момент е алгоритъмът на McKay, имплементиран в пакета Nauty [34, 35]. От друга страна повечето обекти имат естествено представяне като двоични матрици - структури на инцидентност, блок дизайни, проективни равнини и др. Подобен подход вече е прилаган за класификация на линейни кодове [12], самодуални кодове [8, 9], адамарови матрици [11]. Алгоритъм за тестване на изоморфизъм на двоични матрици е включен в пакета Q-Extension [6].

Проблемът за изоморфизъм на графи може лесно да бъде представен като изоморфизъм на двоични матрици, тъй като тези две комбинаторни структури имат естествено представяне една в друга. Затова всеки комбинаторен обект, който може да бъде представен като граф, може да бъде представен също и като двоична матрица и обратно. В някои случаи представянето като матрици е много по-подходящо. Един скорошен пример за използване на двоични матрици под формата на двуделен граф е разгледан и включен в алгоритъм за генериране на регулярни ориентирани графи с отхвърляне на изоморфни

обекти [5].

Генериране с отхвърляне на изоморфните обекти

Алгоритмите за генериране с отхвърляне на изоморфните обекти обединяват в себе си решение едновременно на задачата за конструиране и задачата за класификация. Основните стратегии, които се използват при разработка на такива алгоритми са чрез запис на еквивалентните, чрез канонична форма (orderly generation) и чрез канонично разширяване (canonical augmentation). Стратегиите са подробно описани от Kaski и Östergård [25]. При първата стратегия се поддържа списък на всички нееквивалентни обекти и новополученият обект се тества за еквивалентност с всеки обект от този списък. Ако е еквивалентен на някой от тях, то той бива отхвърлян. При втората стратегия се конструират и разглеждат само обектите, които са в канонична форма спрямо подходящо избрано канонично изображение. При методите с канонично разширяване обектите се конструират по такъв начин, при който се гарантира, че за даден обект няма да бъдат конструирани еквивалентни на него обекти на други стъпка в алгоритъма. С други думи, прилага се родителски тест за всеки конструиран обект. Терминът „родителски“ произхожда от термините на кореново дърво, с което подходящо се моделира реализирането както на тези три стратегии, така и при пълно изчерпване.

Всички обекти, разглеждани в процеса на търсене са елементи на множеството Ω, наречено домейн. Дърво на търсене наричаме кореново дърво с корен $R \in \Omega$, в което два възела са свързани с ребро, когато единият се получава от другият само с една стъпка на търсене, която се дефинира с правилото $X \mapsto C(X)$. Обектите от множеството $C(X)$, които се конструират с една стъпка на търсене от обект $X \in \Omega$ се наричат негови наследници. При пълно изчерпване се обхожда цялото дърво, а при стратегиите за генериране с отхвърляне на изоморфните обекти не се обхождат клоните на дървото, които водят до получаване на еквивалентни на вече конструирани комбинаторни структури.

Конструиране и класификация на самоортогонални и самодуални кодове

Класификацията на всички двоични самодуални кодове на дадена дължина е един от най-важните проблеми в конструктивната теория на кодирането, поради това, че те имат добри алгебрични свойства и близка връзка с комбинаторни структури като блок дизайни, адамарови матрици и графи.

Съществуват различни връзки между самодуални и самоортогонални кодове и комбинаторни дизайни. Повечето конструкции на кодове от дизайни са базиране на разширяване по блокове. Съществува възможна конструкция

на самодуални кодове от адамарови матрици и адамарови дизайни, които са описани в [40]. Ние разглеждаме подобна на тази конструкция, но в по-общ случай, когато се използват несиметрични дизайни. По този начин се получават самоортогонални кодове. Самоортогоналните кодове са изучавани основно за конструиране на самодуални кодове, но съществуват резултати и за самостоятелното им класифициране [10, 37]. В тази работа разглеждаме специфични двоични самоортогонални кодове, които са свързани с параметри на комбинаторни дизайни.

Най-ранните изследвания на самодуални кодове са от 70-те години на миналия век, когато Vera Pless дава класификация на самодуалните кодове за дължини $n \leq 20$ [37]. След това, отново Vera Pless заедно с Conway и Sloane класифицират самодуалните кодове с дължини $n \leq 30$ [14, 15, 36, 38]. Класификацията на самодуалните кодове с дължина 32 е дадена от Bilous и Van Rees [4], а тези с дължина 34 са класифицирани от Bilous [3]. По-нататък, Melchor и Gaborit класифицират оптималните самодуални кодове с дължина 36, т.е. самодуалните $[36, 18, 8]$ кодове [1]. Harada и Munemasa завършват класификацията на всички кодове с дължина 36 [20] и създават база от данни на всички самодуални кодове, класифицирани до този момент [21].

Основните използвани методи за горепосочените дължини са базирани на т. нар. „building up“ конструкция, при която от кодове с по-малка дължина се получават кодовете с по-големи дължини. [8, 22].

Съществува формула, наречена MAS формула, която дава броя на всички самодуални кодове (включително еквивалентните) с дължина n, а именно $N(n) = \prod_{i=1}^{n/2-1}(2^i + 1)$.

Броят на нееквивалентните кодове расте експоненциално спрямо дължината им. След класифицирането на всички самодуални кодове с дължина 36, задачата изглежда непосилна за по-големи дължини. Но разработването на нов метод, базиран на идеята за генериране с отхвърляне на изоморфните обекти чрез канонично разширяване, прави възможна и класификацията на всички самодуални кодове с дължина 38, а също и на оптималните самодуални кодове с дължина 40 [2, 8, 9, 21]. Конструкцията в този алгоритъм е подобна на предходните конструкции, но е комбинирана с родителски тест (а не с тест за еквивалентност на кодове). Частична класификация на самодуалните кодове с дължина 40 е направена и от Betsumiya, Harada и Munemasa, които дават броя на нееквивалентните двойночетни самодуални кодове с дължина 40 [2].

В тази работа е представен алгоритъм за класифициране на самодуални кодове с минимално разстояние 4, който е част от класификацията на всички двоични самодуални кодове с дължина 40. Повече от 70 процента от всички нееквивалентни самодуални кодове с дължина $n = 36$ и $n = 38$ имат мини-

мално разстояние 4. Техните групи от автоморфизми са нетривиални и родителският тест би бил тежък процес, поради което отделната им класификация многократно намалява сложността на цялата класификация. Задачата за класификация на всички самодуални $[40, 20]_2$ кодове е разделена на две подзадачи - класификация на кодовете с минимално разстояние ≥ 6 и класификация на всички самодуални $[40, 20, 4]$ кодове. Случаят при $d = 2$ е тривиален - броят на самодуалните $[40, 20, 2]$ кодове съвпада с броя на всички нееквивалентни самодуални $[38, 19]$ кодове. За конструирането на всички самодуални $[40, 20, 4]$ кодове се използват кодовете с дължина 36 и конструкция, при която се добавят четири координати. Алгоритъмът за класификация на кодовете с минимално разстояние 6 и 8 е от същия тип като този представен в [8]. Резултатът от съчетанието на тези два алгоритъма е класифицирането на всички двоични самодуални кодове с дължина 40, които са 8 250 058 081 на брой.

Обзор на дисертацията

Изследванията са структурирани по следния начин: Дисертацията се състои от увод и 4 глави.

В **Първа глава** са дадени основните дефиниции за линейни кодове. Посочени са някои известни конструкции на самоортогонални и самодуални кодове и постигнатите до момента резултати в изследванията върху тях. Също така са представени основните концепции на алгоритмите за генериране с отхвърляне на изоморфните обекти, базирани на канонична форма и канонично разширяване, на които се основават представените в дисертацията алгоритми.

Във **Втора глава** е представен проблемът за изоморфизъм на комбинаторни обекти чрез изоморфизъм на графи и двоични матрици.

В Раздел 2.1 е представен изоморфизмът на двоични матрици. Двоична матрица наричаме всяка матрица, чиито елементи са от азбуката $\mathbb{F}_2 = \{0, 1\}$. Две двоични матрици A и B от една и съща размерност са изоморфни ($A \cong B$), ако редовете на A могат да се бъдат получени от редовете на B чрез пермутация на колоните на B. Аналогично можем да кажем, че матриците A и B са изоморфни, ако едната може да се получи от другата чрез пермутации на редове и стълбове. Всички изоморфизми между две матрици образуват множеството $Iso(A, B)$.

Комбинаторните структури имат специфични дефиниции за еквивалентност, в които са дадени възможните трансформации за получаване на един обект от друг. Поради това представянето на конкретна структура чрез двоична матрица трябва да бъде внимателно съобразено с тях. Това налага да се въведе оцветяване на матриците по редове (или по стълбове). Оцветяване на матрица $A \in \Omega$ наричаме функцията $\pi_A : A_c \to \mathbb{Z}$, където A_c е множеството от стълбове на A. Цялото число $\pi_A(v)$ за $v \in A_c$ е цветът на стълба v.

Ако $c_1 < c_2 < \cdots < c_s$ са различни цветове на стълбове от A, $s \leq n$, векторът $c = (c_1, c_2, \ldots, c_s) \in \mathbb{Z}^s$ наричаме вектор на цветовете на A. Оцветяването на матрицата A дефинира наредено разбиване на стълбовете и, което означаваме с $\pi = (V_1, V_2, \ldots, V_s)$, а самата оцветена матрица означаваме с наредената тройка (A, π, c). Две оцветени матрици (A, π, c) и (B, σ, d) с еднакви размери са изоморфни, ако съществува пермутация $p \in Iso(A, B)$, която изобразява стълбове от един цвят в стълбове от същия цвят. Аналогично се представя и оцветяването на матрица по редове. Случаят, когато е необходимо оцветяване и по редове и по стълбове може да се сведе до оцветяване само на редове (стълбове) чрез подходящо разширяване на матрицата.

В Раздел 2.2 е представен проблемът за изоморфизъм на графи. Краен неориентиран граф наричаме двойката $G(V, E)$, където $V = \{v_1, v_2, \ldots, v_n\}$ е множеството от върхове на графа, а $E = \{e_1, e_2, \ldots, e_m\}$ е множество от двуелементни подмножества на V ($e_i = \{v_x, v_y\}$), които наричаме ребра на графа. Върховете, които са свързани с ребро наричаме съседни. Два графа, G и H, наричаме изоморфни, ако съществува биекция $f : V(G) \to V(H)$ такава, че за всеки два върха $u, v \in V(G)$ $\{u, v\} \in E(G)$ тогава и само тогава, когато $\{f(u), f(v)\} \in E(H)$, за $\forall u, v \in V(G)$. Биекция f наричаме изоморфизъм между G и H. Множеството от всички изоморфизми между G и H означаваме с $Iso(G, H)$. Изоморфизъм на графа G върху себе си наричаме автоморфизъм.

Съществува естествена връзка между графи и двоични матрици. Всяка двоична матрица може да бъде представена като оцветен двуделен граф. Двуделен граф наричаме графът G с множество от върхове $V(G) = V_1 \bigcup V_2$, където $V_1 \bigcap V_2 = \emptyset$. Ако $\{v_i, v_j\} \in E(G)$, то $v_i \in V_1, v_j \in V_2$.

Ако означим редовете и стълбовете на $m \times n$ матрицата A съответно с $a_1, a_2, \ldots, a_m$ и $b_1, b_2, \ldots, b_n$, можем да конструираме двуделен граф $G = (V_1, V_2, E)$ с множество от върхове $V = V_1 \bigcup V_2$, където $V_1 = \{a_1, a_2, \ldots, a_m\}$, $V_2 = \{b_1, b_2, \ldots, b_n\}$, а E се състои от всички двойки $\{a_i, b_j\}$, за които $A_{ij} = 1$.

Обратно, нека е даден двуделен граф $G = (V_1, V_2, E)$, $V_1 = \{a_1, a_2, \ldots, a_m\}$, $V_2 = \{b_1, b_2, \ldots, b_n\}$. Можем да съпоставим $|V_1| \times |V_2|$ двоична матрица A^G с елементи $A^G_{ij} = 1$, когато $\{a_i, b_j\} \in E$ и $A^G_{ij} = 0$ в противен случай. Освен това всеки граф може да бъде трансформиран до двуделен.

За да бъде записан двуделен граф (двоична матрица, получена от граф) са необходими $(m+n)^2$ клетки компютърна памет. Ние представяме метод, при който граф G с n на брой върха е представен чрез $2n \times 2n$ оцветена двоична матрица G_b, която се представя също с $2n \times 2n$ клетки в паметта. Два графа са изоморфни ако съответните им оцветени двоични матрици са изоморфни. При изоморфизъм на ориентирани графи се използва същия метод. Подобно е и представянето на нелинейни кодове.

Проблемите за изоморфизъм на линейни и нелинейни кодове са разгледани в Раздел 2.3. Дава се представяне чрез двоични матрици, а също така е дадена и връзката на линейни кодове с мултимножества от точки в проективна геометрия, отново посредством оцветени двоични матрици.

Нелинеен q-ичен (n, M) код C наричаме множество от M думи с дължина n над азбука A с q елемента. Два нелинейни кода C_1 и C_2 са еквивалентни, ако единият може да се получи от другият чрез последователно прилагане на пермутации на координатите на кода и пермутации на стойностите във всяка координата.

Нека кодът C е зададен чрез $M \times n$ матрица, чиито редове са кодовите

му думи. На всеки елемент от азбуката $A = \mathbb{Z}_q$ съпоставяме вектор с дължина q по следния начин: $0 \mapsto (10\ldots0)$, $1 \mapsto (010\ldots0)$, $\ldots$, $q-1 \mapsto (0\ldots01)$. След това разширяваме матрицата с още n реда, оцветени в различен цвят, които маркират стълбовете, представляващи една координата. Получаваме двоична матрица C_b.

Пример 0.1. Нека C е $(4,2,3)$ троичен нелинеен код. Чрез горепосочените трансформации получаваме двоичната матрица C_b.

$$C = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 2 \end{pmatrix} \quad C_b = \left(\begin{array}{cccc} 100 & 010 & 010 & 001 \\ 010 & 100 & 001 & 001 \\ \hline 111 & 000 & 000 & 000 \\ 000 & 111 & 000 & 000 \\ 000 & 000 & 111 & 000 \\ 000 & 000 & 000 & 111 \end{array}\right) \begin{array}{c} r_1 \\ r_1 \\ r_2 \\ r_2 \\ r_2 \\ r_2 \end{array}$$

Два нелинейни кода са еквиваленти, ако съответните им оцветени двоични матрици са изоморфни.

При линейни кодове конструирането на двоична матрица е по-сложно. Съществуват три възможни трансформации между еквивалентни кодове - пермутация на координатни позиции, умножение на елементите на дадена координатна позиция с ненулев елемент на полето $\mathbb{F}_q$ и прилагане на автоморфизъм на полето върху елементите във всички координатни позиции.

За да представим линеен код C чрез двоична матрица използваме негово подмножество $\mathcal{B}$, което да е стабилно относно действието на $Aut(C)$. избираме множеството $\mathcal{B}$ да бъде равно на $B_{d_1} \cup B_{d_2} \cup \cdots \cup B_{d_t}$, където B_i е множеството от всички кодови думи в C с тегло i, като $d \leq i \leq n$.

$\mathcal{B}$ има следните свойства:

1. $d = d_1 < d_2 < \cdots < d_t \leq n$,
2. $B_i = \emptyset$ за $d_j < i < d_{j+1}$, $j = 1, \ldots, t-1$,
3. $\mathcal{B}$ генерира C като векторно пространство, но $\mathcal{B} \setminus B_{d_t}$ не генерира кода.

Практически се търси минималното такова множество.

Нека C е линеен код над полето $\mathbb{F}_q$ и $\mathcal{B}$ е негово подмножество, стабилно относно действието на групата от автоморфизми. Нека е матрица, чиито редове са думите от $\mathcal{B}$. Нека $q = p^m$, където p е характеристиката на полето, и нека α е примитивен елемент на $\mathbb{F}_q$. Съпоставяме на всеки ненулев елемент α^j на полето, $0 \leq j \leq q-2$, $2(q-1) \times 2(q-1)$ двоична матрица, с която заместваме всеки

един от елементите на матрицата . Добавят се допълнителни редове, които да гарантират запазването на координатите при прилагане на пермутации, както при представянето на нелинейни кодове и ориентирани графи.

В последния четвърти раздел от Втора глава се дава представяне на адамарови матрици чрез двоични матрици. Адамарова матрица H от ред n наричаме $n \times n$ матрица с елементи ± 1, удовлетворяваща равенството $HH^t = nI$. Две адамарови матрици H_1 и H_2 са (Адамар) еквивалентни, ако H_2 може да бъде получена от H_1 чрез последователност от пермутации на редове и стълбове, умножение на редове и стълбове с -1. Тези трансформации трябва да имат аналогични на тях трансформации и върху конструираната двоична матрица. На всяко 1 и -1 от адамаровата матрицата H съпоставяме матрици както следва:

$$1 \to \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad -1 \to \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

По този начин получаваме двоична матрица с размерност $2n \times 2n$, в която редовете и стълбовете са групирани по двойки. Две адамарови матрици H_1 и H_2 са еквивалентни тогава и само тогава, когато двоични матрици H_{1b} и H_{2b} са изоморфни.

Коментари

Представените методи са разработени съвместно с И. Буюклиев, докладвани са на международна конференция по приложения на компютърната алгебра (ACA) през 2012 година, която бе проведена в София, и са приети за публикуване в Serdica Journal of Computing [P3]. Статията е обзорна и има за цел да представи проблемът за изоморфизъм на комбинаторни обекти и как той може да бъде редуциран до изоморфизъм на основни обекти.

Първоначален вариант на метода за представяне на изоморфизъм на линейни кодове чрез двоични матрици е представен в [7] и е използван в алгоритъм за класификация на самодуални кодове. Тук е даден обновен метод, който е приложим и за линейни кодове над съставни полета. Докладван е на националния семинар по теория на кодирането през 2012 година.

В **Трета глава** разглеждаме метод за конструиране на самоортогонални кодове от комбинаторни дизайни.

Балансиран непълен блок дизайн (Balanced incomplete block design, BIBD) [13] с параметри $2-(v,b,r,k,\lambda)$, или $2-(v,k,\lambda)$, наричаме двойката (V,B), където V е множество с мощност v, чиито елементи се наричат точки, а B е фамилия от b k-елементни подмножества на V, чиито елементи наричаме блокове, така, че всяка точка се съдържа точно в r блока и всеки две точки се

съдържат точно в λ блока. Връзката между дизайни и самоортогонални кодове е разгледана в раздел 3.1.

Дизайните се представят еднозначно чрез матрици на инцидентност - $v \times b$ матрицата A с елементи $a_{ij}, 1 \leq i \leq v, 1 \leq j \leq b$, такива че $a_{ij} = 1$ ако $v_i \in B_j$ и $a_{ij} = 0$ в противен случай.

Самоортогонални кодове конструираме от матрици, наречени IMD (incidence matrix of a design) матрици, които имат свойствата на матрица на инцидентност на дизайн, но с по-малък брой редове. Връзката се дава от следната теорема:

Теорема 0.1. *Нека A е (v, k, λ) IMD матрица с размерност $v' \times b$, $v' \leq v$, и $r - \lambda$ е нечетно цяло число. Тогава:*

1. *ако r е нечетно цяло число, тогава матрицата $(I\ A)$ е пораждаща матрица на самоортогонален код C с дължина $v' + b$. Ако $r \equiv 3 \pmod 4$, то кодът C е двойно четен.*

2. *ако r е четно цяло число и b е нечетно цяло число, то $(I\ \tilde{A})$ е пораждаща матрица на самоортогонален код C с дължина $v' + b + 2$. Ако $r \equiv 2 \pmod 4$ и $b \equiv 3 \pmod 4$, то кодът C е двойно четен.*

В раздел 3.1 е представен алгоритъмът за конструиране на IMD матрици, който е основан на идеята за канонична форма. Матриците се генерират ред по ред, като за област на търсене е взето множеството $\Omega = |\{u | u \in \mathbb{F}_2^b, wt(u) = r\}|$, където b и r са предварително зададени параметри (съответни на параметри на комбинаторен дизайн). При всяко разширяване се проверява по евристичен начин дали получената матрица е в канонична форма.

За каноничен представител на двоични матрици с определена размерност взимаме матрица в тотална лексикографска форма. Нека A бъде $n \times m$ двоична матрица и $G = S_n \times S_m$ бъде групата от всички пермутации на нейните редове и колони. Означаваме редовете на A с $a_1 = (a_{11}\ a_{12}\ \ldots\ a_{1m}), a_2 = (a_{21}\ a_{22}\ \ldots\ a_{2m}), \ldots, a_n = (a_{n1}\ a_{n2}\ \ldots\ a_{nm})$. Конструираме вектор v_A с дължина $n.m$ от редовете на матрицата по следния начин:

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}$$

$$v_A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} & a_{21} & a_{22} & \ldots & a_{2m} & \ldots & a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}$$

Дефиниция 0.1. *Тотална лексикографска форма на дадена матрица A е матрицата A_{tl}, такава, че $v_{A_{tl}} \succ v_{g(A)}$, $\forall g \in G$.*

Всеки клас на еквивалентност съдържа само една матрица в тотална лексикографска форма. Тази матрица е сортирана по редове и по стълбове. Проверката за каноничност се състои в това дали получената сортирана матрица има лексикографски по-голям вектор от еквивалентните на нея сортирани матрици. Казваме, че е търсенето е евристично, тъй като не се генерират всички еквивалентни сортирани матрици. Допускаме възможността някоя матрица в дървото на търсене да не е в канонична форма. В този случай нейните наследници също няма да са в канонична форма и с голяма вероятност те ще бъдат отхвърлени. Поради това се прилага тест за еквивалентност над крайното множеството от генерирани обекти. Тъй като те са малък процент, допускането на еквивалентни обекти и прилагането на тест за еквивалентност би било по-ефективно от изчерпващо търсене за канонична форма на всяка стъпка.

В раздел 3.3 са представени таблици с получените чрез алгоритъма резултати. В много от случаите се получават оптимални самоортогонални кодове. Всички известни самоортогонални кодове са получени за параметрите $[15, 5]$, $[16, 6]$, $[17, 6]$, $[19, 8]$, $[20, 9]$, $[21, 10]$, $[22, 4]$, $[23, 5]$, $[24, 6]$ [10], но в някои други случаи са получени съвсем малък брой - $[34, 4]$, $[35, 5]$, $[36, 6]$. Освен това са конструирани и кодове, които не са посочени в [10] - 82 $[26, 8, 8]$ кода, 18 $[27, 9, 8]$ кода, 4 [28,10,8] кода и 19936 [39,6,16] кода. В повечето случаи броят на получените нееквивалентни самоортогонални кодове е близък до броя на неизоморфните IMD-матрици, но също така има и случаи, като $(19, 10, 5)$, в които броят на IMD-матриците е много по-голям от броя на конструираните самоортогонални кодове. Съществуват параметри на IMD-матрици, от които да се конструират нови линейни кодове [19].

Коментари Представеният алгоритъм е разработен съвместно с И. Буюклиев и В. Монев. Освен параметрите на дизайна на всяка стъпка се прави проверка и за минимално разстояние чрез евристичен алгоритъм на Canteaut и Chabaud, реализиран от В. Монев.

Предварителни резултати са докладвани на Десетата международна конференция "Крайни полета и техните приложения, F_q10", която бе проведена през 2011 година в Гент, Белгия.

Описаният алгоритъм и дадените в дисертацията резултати са публикувани в Problems of Information Transmission [P1].

В **Четвърта глава** е представен методът за класификация на самодуални кодове с дължина 40 и по-точно алгоритъм за конструиране и класифициране

на самодуални кодове с минимално разстояние 4. Този алгоритъм отново е от типа генериране с отхвърляне на изоморфните и се базира на идеята за канонично разширяване, с тази разлика, че не е рекурсивен. Взимайки нееквивалентните самодуални кодове с дължина n получаваме всички нееквивалентни кодове с дължина $n+4$ и минимално разстояние 4.

В първия раздел е дадена теоретичната база на използваната конструкция на самодуални кодове. Основната идея се състои в разширяване на кода (неговата пораждаща матрица) с четири координати (четири стълба). Начинът на разширяване се дава от следната теорема:

Теорема 0.2. *Нека C е двоичен самодуален $[n, k = n/2, 4]$ код и $x = (110\ldots011)$ е кодова дума с тегло 4. Тогава C има пораждаща матрица във вида:*

$$G = \begin{pmatrix} 11 & 00\cdots0 & 00\cdots0 & 1 & 1 \\ 01 & 00\cdots0 & v & 0 & 1 \\ 00 & I_{k-2} & A & a^T & a^T \end{pmatrix}$$

където a и v са двоични вектори с дължина $k-2$. Матрицата $(I_{k-2}|A)$ генерира самодуален $[n-4, n/2-2]$ код C_1.

Нека разгледаме групата от автоморфизми $\mathrm{Aut}(C_1)$ на самодуален $[n-4, n/2-2]$ код C_1 от горната теорема, и нека G_1 е пораждаща матрица на този код. Доказва се, че ако векторите a и b от $\mathbb{F}_2^{k-2}$ принадлежат на една и съща орбита при действие на $Im(f)$ върху $\mathbb{F}_2^{k-2}$, тогава матриците $(G_1\ a^T\ a^T)$ и $(G_1\ b^T\ b^T)$ пораждат еквивалентни кодове.

Взимайки по един вектор от всяка орбита, получаваме нееквивалентни наследници на всеки код, който разширяваме. За да избегнем генериране на еквивалентни кодове от различни родителски възли в дървото на търсене прилагаме родителски тест, който е описан заедно със самия алгоритъм в раздел 4.2. Ако $\overline{B}_1$ и $\overline{B}_2$ са два еквивалентни самодуални $[2k, k, 4]$ кода, които преминават родителския тест, тогава самодуалните $[2k-4, k-2]$ кодове B_1 и B_2 са също еквивалентни. В предишния алгоритъм родителският тест се прави само върху две координатни позиции. В този случай има четири нови координати, които се добавят при конструирането на кода. Тези координати са разделени на две двойки - първите две координати $\{1, 2\}$ и последните две координати $\{n, n-1\}$. Основното е, че двойките трябва да бъдат запазени, когато се прави родителският тест. Първите (последните) две координати трябва да бъдат изобразени или в последните (първите), или да запазят местата си в каноничната форма на кода.

Тъй като родителският тест се прилага при всяко разширяване, бързодействието му е от изключителна важност. Търсене на канонична форма на

линеен код е тежка задача, но може да бъде избегнато чрез подходящо избрани инвариантни функции. Инвариантите и тяхното приложение в родителския тест са описани в раздел 4.3. Дадена инвариантa разделя координатите на кода на псевдоорбити, всяка от които съдържа една или повече реални орбити на действието на групата от автоморфизми на кода върху множеството му от координати. Ако двойките нови координати попадат в различни псевдоорбити, то родителският тест не се изпълнява и кодът ще бъде отхвърлен без да бъде изчислявана каноничната му форма.

Инвариантите трябва да бъдат изчислявани върху целия код, а не само върху пораждаща матрица. За да се оптимизират изчисленията, както по време така и по памет, вместо целия код генерираме пораждащо множество на кода, върху което прилагаме инвариантните функции.

Дефиниция 0.2. *Множеството $M(C)$ от кодови думи на кода C наричаме пораждащо за кода, ако:*

- *$M(C)$ поражда кода C като линейно подпространство;*
- *$M(C)$ е стабилно относно действието на $\mathrm{Aut}(C)$;*
- *ако $C' \cong C''$ и $\sigma(C') = C''$, то $\sigma(M(C')) \equiv M(C'')$;*
- *$M(C)$ е минималното множество с тези свойства.*

Генерирането на пораждащо множество на кода се реализира едновременно с прилагането на инварианти. Ясно е, че множеството от кодови думи с фиксирано тегло е инвариантно относно действието на групата от автоморфизми на кода. Затова първоначално вземаме множеството от кодови думи с минимално тегло, което предварително знаем. С помощта на получените инварианти го разбиваме на псевдоорбити. Взимаме тези от тях, в които имаме базис на кода. При необходимост добавяме кодовите думи със следващите по големина тегла.

Благодарение на тази техника в алгоритъма за класификация на самодуални кодове с минимално разстояние $d = 4$ канонична форма е пресмятана за 5 226 244 513 кода от общо 20 614 314 107 разгледани кода. А в алгоритъма за класификация на самодуални кодове с $d > 4$ канонична форма е пресмятана за 6 563 895 920 кода от общо 131 822 097 145 кода.

В последния раздел е описана самата класификация на всички самодуални $[40, 20]$ кодове, които са 8 211 375 898 на брой. От тях 4 329 329 746 имат минимално разстояние 4, а 3 882 046 152 имат минимално разстояние 6 или 8.

Коментари

Теоретичната база и конструкцията на самодуални кодове са разработени съвместно с И. Буюклиев и са докладвани на XIII-тата международна конференция по алгебрична и комбинаторна теория на кодирането (ACCT), проведена в Поморие през 2014 година [P4]. Имплементацията, алгоритъмът и пълната класификация на самодуалните $[40, 20]$ кодове са направени съвместно с И. Буюклиев и В. Монев и са докладвани отново на международна конференция ACCT, проведена в Калининград, Русия, през 2014г [P5], а статията с пълните доказателства и резултати за класификацията на самодуалните кодове с дължина 40 е приета за публикуване в *IEEE Transaction of Information Theory* [P2].

При реализирането на програмите GenSelfDualD4 и GenSelfDualAllD, верификацията на резултатите, обобщаването на получените резултати и интерфейсната част са направени от В. Монев.

И двата алгоритъма, за генериране на самодуални кодове с минимално разстояние четири и за генериране на самодуални кодове с всяко минимално разстояние, включват нова версия написана на C на алгоритъма за еквивалентност на линейни кодове, разработен от Илия Буюклиев [7].

Заключение

В дисертационния труд е изследван проблема за изоморфизъм на различни комбинаторни структури. Във връзка с това са разработени методи за представяне на структурите чрез двоични матрици, чрез които се оптимизира нужното време и компютърна памет при тест за изоморфизъм. Подходите биха могли да бъдат имплементирани в алгоритми за класификация, а също и да се използват при стратегии за генериране с отхвърляне на изоморфните.

При алгоритмите за конструиране на самоортогонални кодове и за класификация на самодуални кодове са използвани тези стратегии. При първият е приложена идеята за канонична форма на IMD матрици, чрез които се конструират самоортогонални кодове със специфични параметри. Алгоритъма може да се използва за конструиране на нови кодове.

Представен е алгоритъм за конструиране на самодуални кодове с минимално разстояние 4, който е част от класификацията на всички самодуални кодове с дължина 40. Във връзка с нея са имплементирани функции за генериране на пораждащо множество на линеен код и функции за изчисляване на инварианти на линеен код. Пораждащо множество на линеен код се използва и при представяне на проблема за изоморфизъм на двоични линейни кодове чрез двоични матрици.

Апробация на резултатите

Резултатите, включени в дисертацията, са получени в съавторство с

- Буюклиев [P3], [P4];
- Буюклиев и Монев [P1], [P2], [P5].

Публикувани са или са приети за публикуване в международни научни списания:

- *Problems of Information Transmission* [P1];
- *IEEE Transactions on Information Theory* [P2];
- *Serdica Journal of Computing* [P3].

Резултати от дисертацията са докладвани на:

- Национален семинар по теория на кодирането 2010-2013г.
- 3th and 4th International Colloquium on Differential Geometry 2010г, 2014г.
- 10th International Conference on Finite Fields and Their Applications F_q10, 2010г.
- 18th International Conference on Applications of Computer Algebra, 2012г.
- 13th and 14th International Workshop on Algebraic and Combinatorial Coding Theory 2012, 2014г.

Авторска справка

По мнение на автора, основните приноси на дисертационния труд са:

- Представяне на проблема за изоморфизъм на комбинаторни обекти, чрез изоморфизъм на двоични матрици:

  (1) Разработен е метод за свеждане на проблема за изоморфизъм на ориентирани и неориентирани графи към изоморфизъм на оцветени двоични матрици;

  (2) Разработен е метод за свеждане на проблема за изоморфизъм на линейни и нелинейни кодове към изоморфизъм на оцветени двоични матрици;

  (3) Разработен е метод за свеждане на проблема за изоморфизъм на адамарови матрици към изоморфизъм на оцветени двоични матрици;

- Конструиране на самоортогонални кодове чрез комбинаторни дизайни:

  (4) Дефинирано е понятието IMD матрица, свързана с параметри на дизайн и е разработена конструкция на самоортогонални кодове от IMD матрици;

  (5) Разработен е алгоритъм за генериране на IMD матрици и комбинаторни дизайни, базиран на идеята за генериране с отхвърляне на изоморфните обекти чрез канонична форма; Представена е канонична форма на двоична матрица, за която е имплементиран евристичен метод на търсене;

  (6) Дадени са таблици с генерираните кодове, които имат пораждащи матрици в систематична форма $G = (IA)$, където матрицата A е IMD матрица;

  (7) Направен е анализ на получените резултати;

- Алгоритъм за конструиране на самодуални кодове с минимално разстояние $d = 4$:

  (8) Разработена е нерекурсивна конструкция за самодуални $[2k + 4, k + 2, 4]$ кодове от самодуални $[2k, k]$ кодове, която е имплементирана в алгоритъм за класификация на самодуални кодове с минимално разстояние 4. Алгоритъмът е основан на идеята за генериране с отхвърляне на изоморфните чрез канонично разширяване;

(9) Разработени са и са имплементирани инвариантни функции на линеен код, свързани с родителския тест в алгоритъма;

(10) Представен е метод за генериране на пораждащо множество на линеен код, което се използва както при изчисляване на инварианти, така и при тест за изоморфизъм на двоични линейни кодове;

(11) Представена е класификацията на всички самодуални кодове с дължина 40.

Благодарности

Бих искала да изразя своята голяма благодарност към научния си ръководител проф. дмн Илия Буюклиев за актуалните задачи, за разбирането и за подкрепата през периода на докторантурата и подготовката на този труд. Сърдечно благодаря на проф. дмн Стефка Буюклиева за интереса към работата ми, за помощта и за ценните съвети. Бих искала искрено да благодаря на доц. д-р Валентин Бакоев за това, че ме насочи към научната работа. Благодарна съм му за търпението и помощта в преподавателската ми дейност.

Изказвам благодарност на колегите от секция „Математически основи на информатиката“ към ИМИ-БАН за тяхното съдействие и отзивчивост. Бих искала да благодаря и на колегите си от катедра „Алгебра и геометрия“ и от катедра „Информационни технологии“ към факултет „Математика и Информатика“ на Великотърновски университет.

Благодаря на съпруга си и на родителите си за съпричастността към работата ми.

Публикации по дисертацията

[P1] M. Dzhumalieva-Stoeva, I. Bouyulkiev, V. Monev, Construction of self-orthogonal codes from combinatorial designs, *Problems of Information Transmissions*, Vol. 48, No. 3, (2012) pp. 250-258.

[P2] I. Bouyulkiev, M. Dzhumalieva-Stoeva, V. Monev, Classification of Binary Self-dual Codes of Length 40, to appear in *IEEE Trans. Info. Theory*, ISSN 0018-9448 (print), ISSN 1557-9654 (web).

[P3] I. Bouyulkiev, M. Dzhumalieva-Stoeva, Representing Equivalence Problems for Combinatorial Objects, to appear in *Serdica Journal of Computing*, ISSN 1312-6555 (print), ISSN 1314-7897 (web).

[P4] I. Bouyukliev, M. Dzhumalieva-Stoeva, On an algorithm for classification of binary self-dual code with minimum distance four, *Proceedings of 13th International Workshop on ACCT*, Pomorie, Bulgaria, (2012) pp.105-110.

[P5] I. Bouyukliev, M. Dzhumalieva-Stoeva, V. Monev, On the Classificaton of the Binary Self-Dual Codes of Length 40, *Proceedings of 14th International Workshop on ACCT*, Kaliningrad, Russia, (2014) pp.97-102.

Л И Т Е Р А Т У Р А

[1] C.Aguilar-Melchor, P.Gaborit, On the classification of extremal [36,18,8] binary self-dual codes, *IEEE Trans. Inform. Theory*, vol. 54, 2008, pp. 4743-4750.

[2] K. Betsumiya, M. Harada and A. Munemasa, A complete classification of doubly even self-dual codes of length 40, *Electronic J. Combin.* vol. 19, 2012, P18 (12 pp.).

[3] R.T. Bilous, Enumeration of the binary self-dual codes of length 34, *J.Combin. Math. Combin. Comput.*, vol. 59, 2006, pp. 173-211.

[4] R.T. Bilous, G.H.J. Van Rees, An enumeration of binary self-dual codes of length 32, *Designs, Codes and Cryptography*, vol. 26, 2002, pp. 61-68.

[5] G. Brinkmann, Generating regular directed graphs, *Discrete Mathematics*, vol. 313, 2013, pp. 1-7.

[6] I. Bouyukliev, What is Q-Extension?, *Serdica J. Computing*, vol. 1, 2007, pp. 115-130.
http://www.moi.math.bas.bg/~ iliya/Q_ext.htm

[7] I. Bouyukliev, About the code equivalence, in *Advances in Coding Theory and Cryptology*, T. Shaska, W.C. Huffman, D. Joyner, V. Ustimenko, Series on Coding Theory and Cryptology, World Scientific Publishing, Hackensack, NJ, 2007.

[8] S. Bouyuklieva and I. Bouyukliev, An algorithm for classification of binary self-dual codes, *IEEE Trans. Inform. Theory*, vol. 58, 2012, pp. 3933-3940.

[9] S. Bouyuklieva, I. Bouyukliev and M. Harada, Some extremal self-dual codes and unimodular lattices in dimension 40, *Finite Fields Appl.*, vol. 21, 2013, pp. 67-83.

[10] I. Bouyukliev, S. Bouyuklieva, T.A. Gulliver, P. Ostergard, Classification of Optimal Binary Self-Orthogonal Codes, *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 59, 2006, pp. 33-87.

[11] I. Bouyukliev, V. Fack, J. Winne, 2-(31,15,7), 2-(35,17,8) and 2-(36,15,6) designs with automorphisms of odd prime order, and their related Hadamard matrices and codes, Designes, Codes and Cryptography, vol. 51, no.2, 2009, pp. 105-122.

[12] I. Bouyukliev and J. Simonis, Some new results on optimal codes over F_5, *Designs, Codes and Cryptography*, vol. 30, 2003, pp. 97-111.

[13] C. Colbourn and J. Dinitz, *The CRC Handbook of Combinatorial designs*, Boca Raton, FL., CRC Press, 7:3-41, 1996.

[14] J. H. Conway, V. Pless, On the enumeration of self-dual codes, *Journ. Combin. Theory*, ser. A, vol. 28, 1980, pp. 26-53.

[15] J. H. Conway, V. Pless, N.J.A.Sloane, The binary self-dual codes of length up to 32: a revised enumeration, *Journ. Combin. Theory*, ser. A, vol. 60, 1992, pp. 183-195.

[16] P. T. Darga, M. H. Liffiton, K. A. Sakallah and I. L. Markov, Exploiting structure in symmetry detection for CNF, Proceedings of the 41st Design Automation Conference, 2004, pp. 530-534.

[17] P. Foggia, C.Sansone, M. Vento, A Performance Comparison of Five Algorithms for Graph Isomorphism, Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, Ischia, 2001.

[18] T. Fuelner, The automorphism grops of linear codes and canonical representatives of their semilinear isometry classes, *AMC*, vol.3, No.4, 2009, pp. 363-383.

[19] M. Grassl, *Bounds on the minimum distance of linear codes and quantum codes*, online available at http://www.codetables.de.

[20] M. Harada and A. Munemasa, Classification of self-dual codes of length 36, *Advances Math. Communications*, vol. 6, 2012, pp. 229-235.

[21] M. Harada and A. Munemasa, Database of Self-Dual Codes, Online available at http://www.math.is.tohoku.ac.jp/ munemasa/selfdualcodes.htm.

[22] W.C.Huffman, On the Classification and enumeration of self-dual codes, *Finite Fields Appl.* vol. 11, 2005, pp. 451-490.

[23] T. Junttila and P. Kaski, Engineering an efficient canonical labeling tool for large and sparse graphs, Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX07), SIAM, 2007, pp. 135-149.

[24] T. Junttila and P. Kaski, Conflict Propagation and Component Recursion for Canonical Labeling, Proceedings of the 1st International ICST Conference on Theory and Practice of Algorithms, TAPAS 2011, Springer.

[25] P. Kaski, P. R.J. Östergård, Classification algorithms for codes and designs. Springer-Verlag, Berlin Heidelberg, 2006.

[26] W. Kocay, On writing isomorphism programs, Computational and Constructive Design Theory (ed. W. D. Wallis), Kluwer, 1996, pp. 135-175.

[27] D. L. Kreher and D. R. Stinson, Combinatorial Algorithms: Generation, Enumeration and Search, CRC Press, 1999.

[28] J. Leon, Computing automorphism groups of error-correcting codes, *IEEE Trans. Inform. Theory*, vol. 28, 1982, pp. 496-511.

[29] J. S. Leon, An algorithm for computing the automorphism group of a Hadamard matrix, *Journal of Combinatorial Theory A27*, 1979, pp. 289-306.

[30] J. L. Lopez-Presa and A. Fernandez Anta, Fast algorithm for graph isomorphism testing, Proceedings of the 8th International Symposium on Experimental Algorithms, 2009, pp. 221-232.

[31] J. L. Lopez-Presa, A. Fernandez Anta and L. Nunez Chiroque, Conauto-2.0: Fast isomorphism testing and automorphism group computation. Preprint 2011. Available at http://arxiv.org/abs/1108.1060

[32] Z. Mateva, Constructing a canonical form of a matrix in several problems about combinatorial designs, *Serdica Journal of Computing*, vol.2, No. 4, 2008, pp. 349-368.

[33] B. McKay, Isomorph-free exhaustive generation, *J. Algorithms*, vol. 26, 1998, pp. 306-324.

[34] B. McKay, Practical graph isomorphism, *Congressus Numerantium*, vol. 30, 1981, pp. 45-87.

[35] B. McKay and A. Piperno, Practical Graph Isomorphism, II, *J. Symbolic Computation*, vol. 60, 2013, pp. 94-112.

[36] V. Pless, The children of the (32,16) doubly even codes, *IEEE Trans. Inform. Theory*, vol. 24, 1978, pp. 738-746.

[37] V.Pless, A classification of self-orthogonal codes over GF(2), *Discrete Mathematics*, vol. 3, 1972, pp. 209-246.

[38] V. Pless, N.J.A. Sloane, On the classification and enumeration of self-dual codes, *Journ. Combin. Theory*, ser. A 18, 1975, pp. 313-335.

[39] N. Sendrier, The Support Splitting Algorithm, *IEEE Trans, Info. Theory*, vol. 46, 2000, pp. 1193-1203.

[40] V. Tonchev, Combinatorial structures and codes, Sofia, 1988.